

AD-A266 818



WL-TR-93-3025



ASTROS ENHANCEMENTS

Volume I — ASTROS USER'S MANUAL

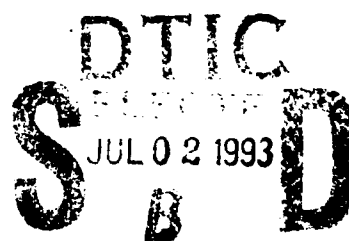
D.J. NEILL
D.L. HERENDEEN

Universal Analytics, Inc.
3625 Del Amo Blvd. Suite 370
Torrance, CA 90503

March 1993

INTERIM REPORT FOR THE PERIOD 1/15/87-10/30/92

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION IS UNLIMITED



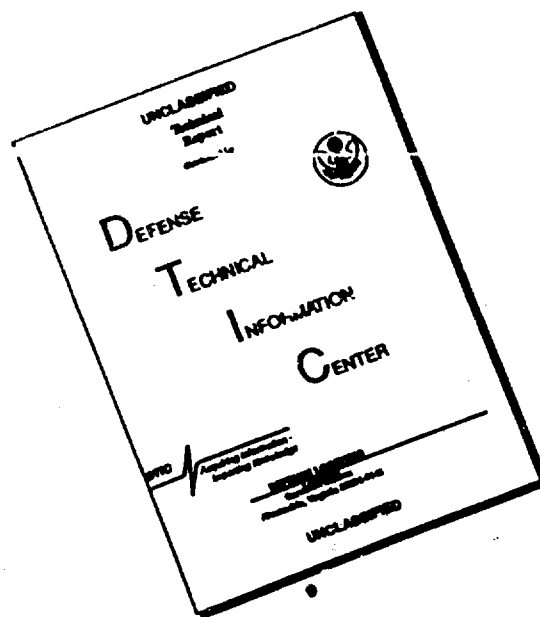
FLIGHT DYNAMICS DIRECTORATE
WRIGHT LABORATORY
AIR FORCE MATERIEL COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OHIO 45433-7562

93-15101



93 7 01 036

DISCLAIMER NOTICE




THIS DOCUMENT IS BEST QUALITY AVAILABLE. THE COPY FURNISHED TO DTIC CONTAINED A SIGNIFICANT NUMBER OF PAGES WHICH DO NOT REPRODUCE LEGIBLY.

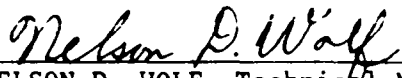
NOTICE

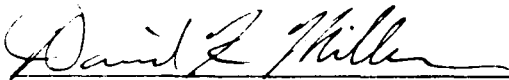
When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely related Government procurement operation, the United States Government thereby incurs no responsibility or any obligation whatsoever. The fact that the Government may have formulated or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication, or otherwise as in any manner, as licensing the holder or any other person or corporation; or as conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

This report is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations.

This technical report has been reviewed and is approved for publication.


STEPHEN M. PITROF, Capt, USAF
Project Engineer
Design & Analysis Methods Section


NELSON D. WOLF, Technical Manager
Design & Analysis Methods Section
Analysis & Optimization Branch


DAVID K. MILLER, Lt Col, USAF
Chief, Analysis & Optimization Branch
Structures Division

"If your address has changed, if you wish to be removed from our mailing list, or if the addressee is no longer employed by your organization please notify WL/FIBRA Bldg 45, 2130 Eighth St Ste 11, Wright-Patterson AFB OH 45433-7552 to help us maintain a current mailing list".

Copies of this report should not be returned unless return is required by security considerations, contractual obligations, or notice on a specific document.

REPORT DOCUMENTATION PAGE			Form Approved OMB No 0704-0188	
<small>This report is prepared for the collection of information. It is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing the burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.</small>				
1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE MAR 1993	3. REPORT TYPE AND DATES COVERED INTERIM 01/15-87--10/30/92	
4. TITLE AND SUBTITLE ASTROS ENHANCEMENTS VOLUME 1 - ASTROS USER'S MANUAL			5. FUNDING NUMBERS CONTR: F33615-87-C-3216 PE: 62201F PROJ: 2401 TASK: 02 WORK UNIT: 82	
6. AUTHOR(S) D. J. NEILL D. L. HERENDEEN				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) UNIVERSAL ANALYTICS INC. 3625 DEL AMO BLVD STE 370 TORRANCE CA 90503			8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) FLIGHT DYNAMICS DIRECTORATE WRIGHT LABORATORY AIR FORCE MATERIEL COMMAND WRIGHT PATTERSON AFB OH 45433-7562			10. SPONSORING / MONITORING AGENCY REPORT NUMBER WL-TR-93-3025	
11. SUPPLEMENTARY NOTES				
12a. DISTRIBUTION / AVAILABILITY STATEMENT APPROVED FOR PUBLIC RELEASE: DISTRIBUTION IS UNLIMITED.			12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words) ASTROS (Automated STRuctural Optimization System) is a computer program for the multidisciplinary design and analysis of aerospace structures. ASTROS combines mathematical optimization algorithms with traditional structural analysis disciplines such as static forces, normal modes, static aeroelasticity, and dynamic aeroelasticity (flutter), all in a finite element context, to perform automated preliminary design of an aircraft structure. This report is a complete user's manual that documents the many features of ASTROS through version 10 of the software package. It also provides information on system architecture and other topics of interest. This report is Volume 1 of a set; Volume 2 (WL-TR-93-3038) is the programmer's manual.				
14. SUBJECT TERMS ASTROS, OPTIMIZATION, AEROELASTICITY, FLUTTER, FINITE ELEMENT METHOD, STRUCTURAL DESIGN, AEROSPACE STRUCTURES			15. NUMBER OF PAGES 440	
			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT UNCLASSIFIED	18. SECURITY CLASSIFICATION OF THIS PAGE UNCLASSIFIED	19. SECURITY CLASSIFICATION OF ABSTRACT UNCLASSIFIED	20. LIMITATION OF ABSTRACT UL	

FOREWORD

This interim report is submitted in fulfillment of CDRL CLIN 0001, Sequence No. 13. of the ASTROS Enhancements Contract, F33615-87-C-3216, dated 29 January 1987. This volume provides the basic user's documentation for the Automated Structural Optimization System.

This work was performed by Universal Analytics, Inc. and their subcontractor, Northrop Corp. This manual supercedes the original ASTROS User's Manual documented in AFWAL-TR-88-3028, Volume II by D.J. Neill, E.H. Johnson and D.L. Herendeen. The major contributors to this report were D.L. Herendeen, the Program Manager, and D.J. Neill, the Associate Program Manager. E.H. Johnson, previously of Northrop, was a major contributor to the original report.

Capt. S. Pitrof is the Air Force Project Engineer and Dr. V.B. Venkayya the initiator of the effort. This report covers the work performed between 29 January 1987 and 30 October 1992, but also includes updated information valid for Version 10.0 of ASTROS.

DTIC QUALITY INSPECTED 5

Accession For	
NTIS GRA&I	<input checked="checked" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By	
Distribution/	
Availability Codes	
Dist	AVAIL CODES
A1	Special

CONTENTS

1. THE ASTROS INPUT DATA	1
1.1. INTRODUCTION	1
1.2. THE INPUT DATA STREAM	3
1.3. THE ASSIGN DATABASE ENTRY — GENERAL	7
1.3.1. ASSIGN DATABASE DESCRIPTIONS FOR HOST COMPUTERS	9
1.3.1.1. VMS IMPLEMENTATION	9
1.3.1.2. IBM MVS IMPLEMENTATION	11
1.3.1.3. UNIX SYSTEM IMPLEMENTATION	12
1.4. THE INCLUDE DIRECTIVE	14
1.5. THE DEBUG PACKET	15
1.5.1. EXECUTIVE SYSTEM DEBUG COMMANDS	16
1.5.2. DATABASE AND MEMORY MANAGER DEBUG COMMANDS	17
1.5.3. INTERMEDIATE RESULTS PRINTING COMMANDS	18
1.5.4. SEQUENCER INTERMEDIATE PRINT COMMANDS	21
2. THE EXECUTIVE SYSTEM AND MAPOL	23
2.1. THE MAPOL PROGRAM	24
2.2. MAPOL EDIT COMMANDS	25
2.3. THE STANDARD EXECUTIVE SEQUENCE	25
2.4. STANDARD EXECUTIVE SEQUENCE STRUCTURE	26
2.4.1. MAPOL Declarations	28
2.4.2. The Solution Algorithm	34
2.4.2.1. MAPOL Engineering and Utility Modules	35
2.4.2.2. The Preface Segment	40
2.4.2.3. The Analysis/Optimization Segments	40
2.4.3. Modifying the Standard MAPOL Sequence	41
2.4.4. Restart Capability	43
2.4.4.1. Introduction	43
2.4.4.2. Ensuring proper STATUS of the run-time database	43
2.4.4.3. Suspending/Restarting Execution	44
2.4.4.4. Resetting MAPOL Parameters	45
2.5. MAPOL PROGRAM LISTING	45

3. THE SOLUTION CONTROL PACKET	99
3.1. TYPE OF BOUNDARY CONDITION	101
3.2. BOUNDARY CONDITION	102
3.3. DISCIPLINES	105
3.3.1. DISCIPLINE OPTIONS	107
3.3.2. STATICS Discipline Options	108
3.3.3. MODES Discipline Options	110
3.3.4. SAERO Discipline Options	110
3.3.5. FLUTTER Discipline Options	110
3.3.6. TRANSIENT Discipline Options	111
3.3.7. FREQUENCY Discipline Options	111
3.3.8. BLAST Discipline Options	111
3.3.9. NPSAERO Discipline Options	112
3.4. OUTPUT REQUESTS	112
3.4.1. Subset Options	113
3.4.2. Response Quantity Options	114
3.4.3. Form Options	117
3.4.4. Labeling Options	117
3.5. SOLUTION CONTROL COMMANDS	118
4. THE BULK DATA PACKET	143
4.1. BULK DATA ECHO OPTIONS	144
4.2. FORMAT OF THE BULK DATA ENTRY	144
4.3. DATA FIELD FORMATS	146
4.4. ERROR CHECKING IN THE INPUT FILE PROCESSOR	146
4.5. BULK DATA ENTRY SUMMARY	147
4.5.1. Aerodynamic Load Transfer	147
4.5.2. Applied Dynamic Loads	147
4.5.3. Applied Static Loads	147
4.5.4. Boundary Condition Constraints	148
4.5.5. Design Constraints	148
4.5.6. Design Variables, Linking and Optimization Parameters	149
4.5.7. Geometry	149
4.5.8. Material Properties	149
4.5.9. Miscellaneous Inputs	150
4.5.10. Output Selection Lists	150
4.5.11. Steady Aerodynamics	150

4.5.12. Structural Element Connection	150
4.5.13. Structural Element Properties	151
4.5.14. Unsteady Aerodynamics	151
4.5.15. Discipline Dependent Problem Control	152
4.6. DIFFERENCES BETWEEN ASTROS AND NASTRAN BULK DATA	152
4.7. BULK DATA DESCRIPTIONS	153
5. ASTROS OUTPUT FEATURES	355
5.1. SYSTEM CONTROLLED OUTPUT	355
5.1.1. Default Output Printed by Modules	358
5.1.2. Error Message Output	360
5.2. SOLUTION CONTROL OUTPUT OPTIONS	362
5.2.1. Element Response Quantities	363
5.2.1.1. Aerodynamic Element Output	364
5.2.1.2. Bar Element Output	364
5.2.1.3. ELAS Element Output	367
5.2.1.4. IHEX1 Element Output	367
5.2.1.5. IHEX2 Element Output	369
5.2.1.6. IHEX3 Element Output	370
5.2.1.7. Rod Element Output	371
5.2.1.8. QDMEM1/TRMEM Element Output	371
5.2.1.9. QUAD4/TRIA3 Element Output	374
5.2.1.10. Shear Panel Output	376
5.2.2. Nodal Response Quantities	376
5.2.3. Design Variables and Design Constraints	380
5.2.4. Flutter/Normal Modes Response Quantities	382
5.2.5. Aeroelastic Trim Quantities	383
5.3. SUMMARY OF SOLUTION RESULTS	387
5.4. OTHER SELECTABLE QUANTITIES	389
5.4.1. Intermediate Steady Aerodynamic Matrix Output	389
5.4.2. Intermediate Unsteady Aerodynamic Matrix Output	389
5.4.3. Flutter Root Iteration Output	390
5.4.4. Stress Constraint Computation Output	390
5.4.5. Intermediate Optimization Output	391
5.5. EXECUTIVE SEQUENCE OUTPUT UTILITIES	391
5.5.1. Structural Set Definition Print Utility, USETPRT	391
5.5.2. Special Matrix Print Utility, UTGPRT	392
5.5.3. General Matrix Print Utility, UTMPRT	392
5.5.4. General Relation Print Utility, UTRPRT	392

5.5.5. General Unstructured Print Utility, UTUPRT	393
5.6. INTERACTIVE CADDDB ENVIRONMENT (ICE)	393
6. MAPOL PROGRAMMING	395
6.1. INTRODUCTION AND USER OPTIONS	395
6.1.1. USER OPTIONS	395
6.1.2. MAPOL PROGRAM FORM	396
6.1.3. THE STANDARD ASTROS SOLUTION	396
6.1.4. MODIFYING THE STANDARD SOLUTION	397
6.1.5. CREATING MAPOL PROGRAMS	397
6.1.6. SUMMARY	397
6.2. DATA TYPES AND DECLARATIONS	398
6.2.1. DEFINITIONS AND NOTATION	398
6.2.2. COMMENTARY	399
6.2.3. SIMPLE DATA TYPES	399
6.2.3.1. Data Type INTEGER	400
6.2.3.2. Data Type REAL	400
6.2.3.3. Data Type COMPLEX	400
6.2.3.4. Data Type LOGICAL	400
6.2.3.5. Data Type LABEL	400
6.2.4. COMPLEX DATA TYPES	400
6.2.4.1. Data Types MATRIX and IMATRIX	401
6.2.4.2. Data Type Relation	401
6.2.4.3. Data Types UNSTRUCT and IUNSTRUCT	403
6.2.4.4. Data Base Entity Declaration Requirements	403
6.3. EXPRESSIONS AND ASSIGNMENTS	403
6.3.1. ARITHMETIC EXPRESSIONS	404
6.3.1.1. Arithmetic Operators	404
6.3.1.2. Arithmetic Operands	404
6.3.1.3. Evaluation of Arithmetic Expressions	405
6.3.1.4. The Uses of Parentheses	405
6.3.1.5. Type and Value of Arithmetic Expressions	406
6.3.2. LOGICAL EXPRESSIONS	406
6.3.2.1. Logical Operators	406
6.3.2.2. Logical Operands	407
6.3.2.3. Evaluation of Logical Expressions	407
6.3.3. RELATIONAL EXPRESSIONS	408
6.3.3.1. Relational Operators	408
6.3.3.2. Relational Operands	409
6.3.3.3. Evaluation of Relational Expressions	409
6.3.4. MATRIX EXPRESSIONS	409
6.3.4.1. Matrix Operators	409

6.3.4.2. Matrix Operands and Expressions	410
6.3.5. ASSIGNMENT STATEMENTS	410
6.4. CONTROL STATEMENTS	411
6.4.1. INTRODUCTION	411
6.4.2. THE UNCONDITIONAL GOTO STATEMENT	411
6.4.3. ITERATION	412
6.4.3.1. The FOR...DO Loop	412
6.4.3.2. The WHILE...DO Loop	413
6.4.4. THE IF STATEMENT	414
6.4.4.1. The Logical IF	414
6.4.4.2. The Block IF	414
6.4.4.3. The IF...THEN...ELSE	415
6.4.4.4. Nested IF Statements	415
6.4.5. THE END AND ENDP STATEMENTS	415
6.5. INPUT/OUTPUT STATEMENTS	415
6.5.1. THE PRINT STATEMENT	416
6.6. PROCEDURES AND FUNCTIONS	416
6.6.1. INTRODUCTION	416
6.6.2. PROGRAM UNITS AND SCOPE OF VARIABLES	416
6.6.3. DEFINING A PROCEDURE	417
6.6.4. INVOKING A PROCEDURE	418
6.6.5. FUNCTION PROCEDURES	419
6.6.5.1. Examples of Variable Scope	419
6.6.6. INTRINSIC FUNCTION PROCEDURES AND INTRINSIC PROCEDURES	419
6.6.7. INTRINSIC MATHEMATICAL FUNCTIONS	420
6.6.8. INTRINSIC RELATIONAL PROCEDURES	421
6.6.9. GENERAL INTRINSIC PROCEDURES	422
7. REFERENCES	425

LIST OF FIGURES

Figure 1. Structure of the ASTROS Input Data Stream	5
Figure 2. Features of a Sample ASTROS Input Stream	6
Figure 3. Function of the ASSIGN DATABASE Entry	8
Figure 4. Structure of the Standard MAPOL Sequence	27
Figure 5. BAR Element Coordinate System	365
Figure 6. BAR Element Forces Sign Conventions	366
Figure 7. IHEX1 Element Geometry	367
Figure 8. IHEX2 Element Geometry	369
Figure 9. IHEX3 Element	370
Figure 10. ROD Element Coordinate System	371
Figure 11. QDMEM1 Element Coordinate System	372
Figure 12. TRMEM Element Coordinate System	373
Figure 13. QUAD4 Element Coordinate System	374
Figure 14. TRIA3 Element Coordinate System	374
Figure 15. Shear Panel Forces	377
Figure 16. Schematic Representation of Relational Data	402
Figure 17. MAPOL Program Using Relational Procedures	422

LIST OF TABLES

Table 1. Command Syntax Conventions	3
Table 2. MAPOL Debug Commands	16
Table 3. Database Debug Commands	17
Table 4. Intermediate Results Debug Commands	18
Table 5. Sequencer Debug Commands	21
Table 6. MAPOL Edit Commands	25
Table 7. Real Parameters in the Standard Sequence	29
Table 8. Integer Modelling Parameters	30
Table 9. Integer Design Parameters	31
Table 10. Integer Aerodynamic Parameters	31
Table 11. Integer Discipline Parameters	32
Table 12. Logical Discipline Parameters	33
Table 13. Summary of ASTROS Modules	35
Table 14. Levels of Solution Control	100
Table 15. Summary of ASTROS Disciplines	106
Table 16. Summary of Discipline Options	109
Table 17. Response Quantities by Discipline	116
Table 18. DEBUG and ASSIGN DATABASE Output	357

Table 19. Boundary Condition Summary	357
Table 20. Active Boundary and Constraint Summary	358
Table 21. Resequencing Summary	358
Table 22. Active Constraint Summary	359
Table 23. Approximate Optimization Summary	359
Table 24. Design Iteration History	360
Table 25. ASTROS Execution Summary	361
Table 26. Aerodynamic and Structural Elements in ASTROS	364
Table 27. BAR Element Output Quantities	366
Table 28. IHEX1 Element Solution Quantities	368
Table 29. ROD Element Solution Quantities	372
Table 30. QDMEM1 Solution Quantities	373
Table 31. QUAD4 and TRIA3 Solution Quantities	375
Table 32. SHEAR Solution Quantities	377
Table 33. Displacement Vector	378
Table 34. Complex Displacement Vector	378
Table 35. Design Variable Values	381
Table 36. Flutter Solution Results	382
Table 38. Real Eigenanalysis Results	383
Table 37. Modal Participation Factors	383
Table 39. Symmetric Trim Results	385

Table 40. Antisymmetric Trim Results	387
Table 41. Summary of Output Quantities	388
Table 42. MAPOL Command Options	396
Table 43. Summary of MAPOL User Options	398
Table 44. MAPOL Arithmetic Operators	404
Table 45. MAPOL Operation Rules	406
Table 46. MAPOL Logical Operators	406
Table 47. Evaluation of MAPOL Logical Expressions	407
Table 48. Relational Operators in MAPOL	408
Table 49. Matrix Operators in MAPOL	409
Table 50. Assignment Rules in MAPOL	411
Table 51. Intrinsic Mathematical Functions in MAPOL	420
Table 52. Intrinsic Relational Procedures in MAPOL	421

1. THE ASTROS INPUT DATA

1.1. INTRODUCTION

This User's Manual is one of four manuals documenting ASTROS (the Automated Structural Optimization System). The other three are the Application Manual, the Programmer's Manual and the Theoretical Manual. The Application and Theoretical Manuals indicate the range of capabilities of the ASTROS system while the Programmer's Manual is provided to give details of the internal workings of the engineering and programming utility modules. The User's Manual provides a complete description of the user interface to the ASTROS system in order to facilitate the preparation of input data. It introduces the features of the ASTROS system that enable the user to direct this software system and documents the mechanisms by which the user can communicate with the system. It is assumed that the reader is familiar, from a study of the Theoretical Manual, with the engineering capabilities of the ASTROS system and is using this manual to define the form of the particular input that directs the system to perform a desired function. This manual is intended to provide the user with a convenient reference for all forms of input to the system and is therefore organized along the same lines as the input data stream. The discussion of each topic is brief and generic and is followed by detailed documentation of the user inputs. Information on ASTROS output formats is in a separate chapter as is the description of the Matrix Analysis Problem Oriented Language (MAPOL) used for programming ASTROS.

This manual is directed toward the engineer/designer/analyst who is using ASTROS to perform engineering design or analysis. While ASTROS is perfectly capable of performing many tasks not explicitly supported in the standard execution, the user must know the engineering software in considerable detail to direct the procedure to perform these alternative functions. The mechanisms by which these more advanced features are invoked are included in this manual but no attempt is made to provide sufficient information to the user to generate new analysis features or to grossly modify the existing capabilities of the system. These more advanced topics are treated in the Programmer's and Application Manuals which document the individual modules in the system and their interactions. Rudimentary (and typical) modifications to the execution sequence and changes to execution parameters are discussed in detail in this manual.

Machine and installation dependent aspects of ASTROS are also contained in the Application and Programmer's Manuals rather than in the User's Manual. Only those machine dependency issues that are logically related to the preparation of the input are discussed in the User's Manual. Machine dependencies in the input are limited to the naming conventions for the run time database file(s) and the parameters that can be used on the `ASSIGN DATABASE` entry. Other machine dependencies are handled as part of the installation of the system on each particular host machine. These issues are documented in the Programmer's Manual since they are relevant only to the "system manager," not to the "user."

It will be apparent to many readers that the NASTRAN structural analysis system was used as a guide in the design of the ASTROS procedure. Both NASTRAN and ASTROS comprise large scale, finite element structural analysis in executive driven software systems. Therefore, many of the input and output features are similar. NASTRAN has become an industry standard in finite element structural analysis with many pre- and post-processors developed around NASTRAN data. In order to facilitate acceptance and use of the ASTROS procedure, NASTRAN compatibility was considered desirable in many areas. As a result, many aspects of the ASTROS input are similar in form or purpose to those in NASTRAN and, in many other cases, the same nomenclature has been adopted. In some instances in this document, therefore, ASTROS input will be compared and contrasted to NASTRAN input in order to present a concise picture of the ASTROS input and to assist the reader familiar with NASTRAN in making the connection to the equivalent item in ASTROS. Although familiarity with NASTRAN is not a prerequisite to understanding the ASTROS documentation, sufficient numbers of potential ASTROS users are expected to be familiar with the NASTRAN system to justify the sometimes casual reference to NASTRAN features.

This Chapter contains a description of the ASTROS input file, database assignment and debug control inputs with Chapters 2 through 4 organized to parallel the input file structure. Within each of these chapters, the function of the particular input packet is presented along with illustrations of how the data are prepared. Each packet is described in a generic fashion so as to indicate how the sophisticated user can make use of the more advanced features of the system without cluttering the discussion with details of the input structures. The detailed documentation of the separate input structures of the data packet then follow within each Chapter. This form of user's documentation enables this manual to be useful as a guide to the beginning user as well as a reference for the experienced user. While there are a number of advanced input features, the required input for most jobs is the ASSIGN DATABASE command, described in Section 1.3, and the Solution Control and Bulk Data packets described in Chapters 3 and 4, respectively.

In Chapter 5, following the input stream descriptions, the output features of the ASTROS system are documented. While these features are selected through directives in the input data stream, they are sufficiently numerous and complex to justify a separate chapter devoted solely to output requests. The output capabilities of the system are described in very general terms while the output requests available for each analysis discipline and optimization feature are documented in detail. Most output is selected through Solution Control directives that are documented in Chapter 3, but some are selected through modifications to the executive (MAPOL) sequence. Chapter 2 documents all of the output utilities available to the user through MAPOL directives and gives several examples of modifying the MAPOL sequence to obtain additional output. Other features are described in the MAPOL Programmer's Manual which comprises Chapter 6.

Many examples of user input are used throughout this document. In order to ease the burden of interpretation, the conventions shown in Table 1 will be used in the examples unless otherwise noted. Chapter 6, which describes the MAPOL programming interface, describes additional conventions required for the programming syntax of MAPOL.

Table 1. Command Syntax Conventions

MAPOL NOGO	Capital letters indicate that the phrase must appear exactly as shown
MAPOL <i>params</i>	Lower case italic symbols act as generic place holders indicating that an option or options can or must be included
MAPOL [GO NOGO]	Symbol(s) enclosed in brackets [] are optional. If more than one symbol is available they will be stacked in vector notation with any defaults denoted by boldface.
INCLUDE <filename>	A required symbol is enclosed in angle brackets. If the angle brackets surround an option list, at least one of the available options must be selected.
BEGIN_BULK	The underscore (_) is used to signify a required blank space.

1.2. THE INPUT DATA STREAM

The ASTROS user directs the system through an input data stream composed of a command to attach the ASTROS run time database followed by multiple data packets. Each packet contains a set of related data providing the information needed to execute the ASTROS procedure. The packets begin with a keyword indicating the nature of the data within the packet and terminate with an ending keyword or with the start of the next data packet. All the packets in the input data stream are optional, although the order in which they must appear is fixed. The purpose of this section is to document the structure of the input data stream. Detailed documentation of the data within each data packet is then presented in separate chapters.

Figure 1 shows the general form of the input data stream and Figure 2 illustrates the actual input stream features with a sample stream for a ten bar truss model. The first non-blank record of the input file must be the **ASSIGN DATABASE** entry. This command enables the user to attach the run time database file(s) that are used during the execution of the ASTROS procedure. There are four optional data packets following the **ASSIGN DATABASE** entry which, if they are present, must appear in the order shown. The first is the **DEBUG** packet which contains *debug* commands to control or select specific actions within the executive and database management systems. The second packet is the **MAPOL** packet containing the executive system control directives consisting of either a standalone MAPOL program or **EDIT** commands to modify the standard MAPOL program. If the MAPOL packet is absent, the unmodified standard MAPOL sequence directs the execution. The Solution Control commands appear in the third optional packet denoted by the keyword **SOLUTION**. These commands select the engineering data to be used in each subcase from the set of data provided in the fourth and final data packet: the

BULK DATA packet. The **BULK DATA** packet contains the engineering data describing the finite element structural model, the aerodynamic model(s), and the design model, as well as all the data needed to perform the specific analysis and/or optimization tasks. The **MAPOL**, **SOLUTION** and **BULK DATA** packets are analogous to the **NASTRAN** executive control, case control and bulk data decks, respectively.

In interpreting the input data stream, **ASTROS** recognizes the keywords shown in Figure 1. These keywords must be the first nonblank characters on the line (leading blanks are allowed) and have the structure shown. In some cases the keyword is also a command line that makes up part of the data packet which it initiates. In these cases, the command parameters are documented in the User's Manual chapter discussing the details of the associated data packet. For example, the **MAPOL** keyword is part of a command that directs the **MAPOL** compiler to take certain actions. The detailed discussion of the **MAPOL** command is therefore contained in Chapters 2 and 6 of this manual. **ASTROS** automatically converts the case of the input data stream when necessary. The only portions of the data which are not converted are file names which are used in the **INCLUDE** and **ASSIGN DATABASE** commands, and the Solution Control commands **TITLE**, **SUBTITLE** and **LABEL**. This allows the user to freely enter data in any case. Be aware, however, that file names are never converted and that when using an **ASTROS** host computer in which case is important, such as Unix, then the correct case must always be used in file names.

Two keyword commands are related only to the input data stream and not to the data within a packet. These are the **ASSIGN DATABASE** and the **INCLUDE** commands. Each of these is discussed in detail in the following sections.

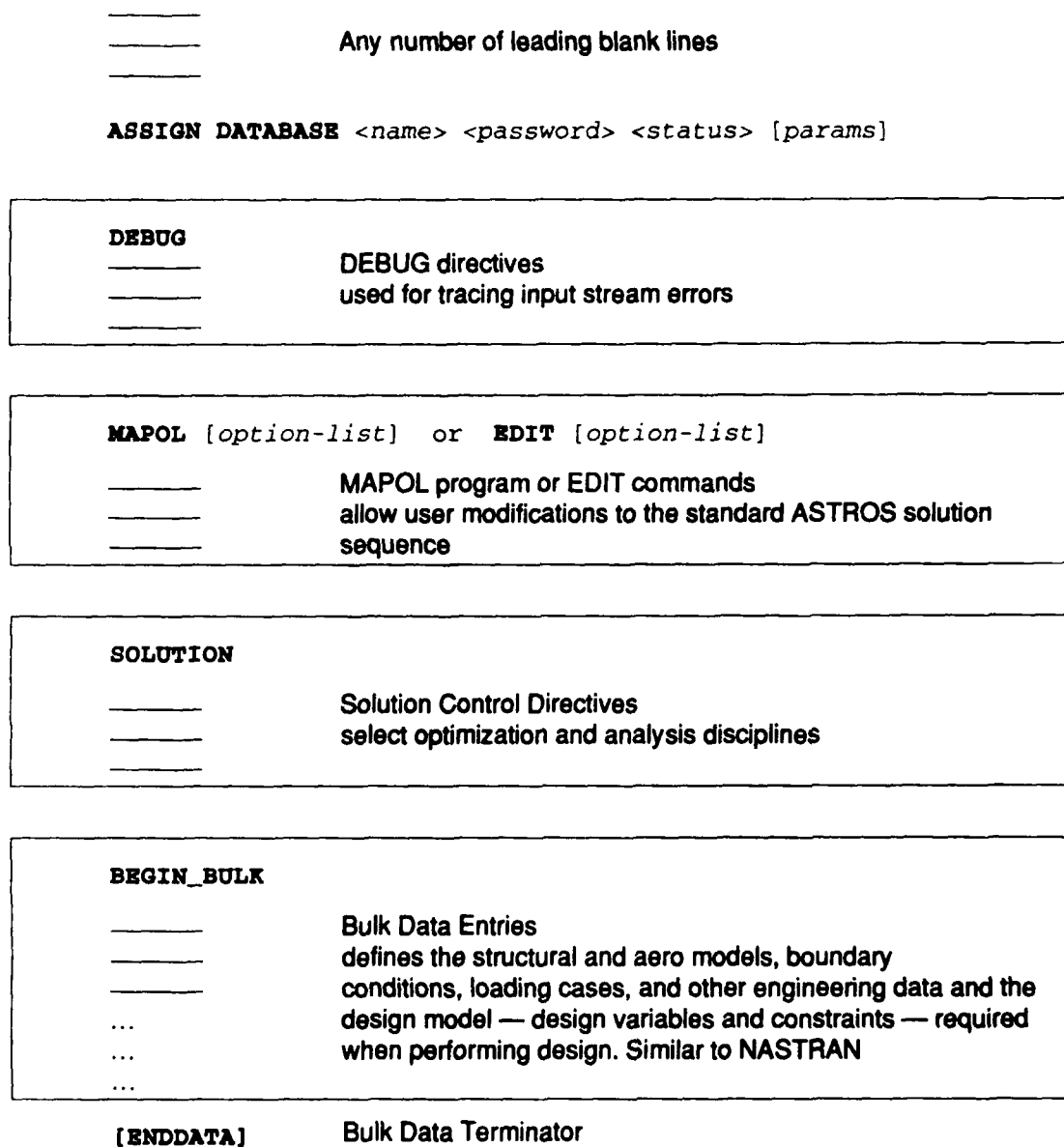


Figure 1. Structure of the ASTROS Input Data Stream

ASSIGN DATABASE TENBAR SHAZAM NEW DELETE

DEBUG

DESIGN=5

EDIT NOLIST

INSERT 1463

CALL UTMPT (, (AMAT));

SOLUTION

TITLE = TEN BAR TRUSS

OPTIMIZE

PRINT DCON=ALL, HIST

BOUNDARY SPC = 1

LABEL = STATIC ANALYSIS

STATICS (MECH = 1), CONST (STRESS = 100, GENERAL = 100)

END

ANALYZE

BOUNDARY SPC = 1, METHOD = 2

STATICS (MECH = 1)

LABEL = FINAL STATIC ANALYSIS

PRINT DISP = ALL

MODES

LABEL = FINAL MODAL ANALYSIS

PRINT (MODES=ALL) DISP = ALL, ROOT=ALL

END

BEGIN BULK

```
GRID, 1, , 720.0, 360.0, 0.0
GRID, 2, , 720.0, 0.0, 0.0
GRID, 3, , 360.0, 360.0, 0.0
GRID, 4, , 360.0, 0.0, 0.0
GRID, 5, , 0.0, 360.0, 0.0
GRID, 6, , 0.0, 0.0, 0.0
CROD, 1, 10, 3, 5
CROD, 2, 10, 1, 3
CROD, 3, 10, 4, 6
...
CROD, 9, 10, 2, 3
CROD, 10, 10, 1, 4
PROD, 10, 2, 15.0
MAT1, 2, 1.E+7, , 0.3, 0.1, , , 25000.0, -25000.0
SPC1, 1, 123456, 5, 6
SPC1, 1, 3456, 1, THRU, 4
FORCE, 1, 2, , -1.E5, 0.0, 1.0, 0.0
FORCE, 1, 4, , -1.E5, 0.0, 1.0, 0.0
CONVERT, MASS, 2.59E-3
EIGR, 2, GIV, 0.0, 700.0, 2, 2, , , ABC, +BC, MAX
MPPARM, ISCAL, 1
DESELM, 1, 1, CROD, 6.667E-3, 1000.0, 2.0, , ROD1
DESELM, 2, 2, CROD, 6.667E-3, 1000.0, 2.0, , ROD2
...
DESELM, 9, 9, CROD, 6.667E-3, 1000.0, 2.0, , ROD9
DESELM, 10, 10, CROD, 6.667E-3, 1000.0, 2.0, , ROD10
DCONSTR, 2, VMISES
DCONVMM, 100, 2.5+4, -2.5+4, , 2
DCONDSP, 100, 1, UPPER, 2.0, POSNOD1, 1, 2, 1.0
DCONDSP, 100, 2, UPPER, 2.0, POSNOD2, 2, 2, 1.0
...
DCONDSP, 100, 8, LOWER, -2.0, NEGNOD1, 4, 2, 1.0
```

ENDDATA

Figure 2. Features of a Sample ASTROS Input Stream

1.3. THE ASSIGN DATABASE ENTRY — GENERAL

The first line of the ASTROS input data stream must be the **ASSIGN DATABASE** entry. This entry identifies the run time database files to be used in the current ASTROS execution and specifies certain parameters associated with the files. The format of this entry is:

```
ASSIGN DATABASE <dbname> <password> <status> [params]
```

where,

dbname	is a name identifying the run time database files (maximum of 8 characters or fewer, depending on the local host. Restricted under IBM/MVS DDNAME syntax to a maximum of 5 characters)
password	is a user assigned password for the database files (maximum of 8 characters)
status	is the status of the database files. Must be either OLD , NEW or TEMP
params	are optional (installation dependent) parameters e.g., DBLKSIZ = n , IBLKSIZ = n , etc.

The entries on the **ASSIGN DATABASE** command are *not* keyword controlled and so must be input in the order shown. They can be separated by any number of blank characters or commas but must reside on no more than 10 physical records of the input data stream. Note that two commas do not imply that a parameter is missing; instead, the second comma will be ignored. Two equivalent examples are shown below:

```
ASSIGN DATABASE, ASTSC, KIMBERLY, OLD  
ASSIGN DATABASE ASTSC KIMBERLY OLD
```

Figure 3 is provided to illustrate the function of the **ASSIGN DATABASE** entry. In the case shown, the installation dependent parameter **VOL** has been implemented which allows selection of the physical device on which the requested file(s) DB1 reside.

The optional **params** on the **ASSIGN DATABASE** entry may or may not be keyword controlled and are installation dependent. They provide a mechanism for the user to direct machine or installation dependent file operations to be performed by the ASTROS procedure. At each site, the installation of the code involves a definition of these parameters and the form they must take. The ASTROS system is currently functional on numerous host systems including IBM 370 series, VAX/VMS, VAX/Ultix, IBM/AIX, SGI 4D series and Crimson/Indigo series, HP/9000 720 and 730 series, CRAY/UNICOS, DEC-Station, Convex, and SunSparcstation. The availability on a specific computer may be obtained on request from the U.S. Air Force.

The next section documents the installation dependent **ASSIGN DATABASE** parameters for some of the more common features/hosts. These features, however, may be customized to a very high degree and may be modified by the local system manager. Further documentation of the **ASSIGN DATABASE** command is left to the local installation or will be included in the delivery material. The Programmer's Manual contains the detailed description of how these and other machine dependent parameters are defined. The following sections are optional. *First-time readers may proceed to Chapters 2 or 3.*

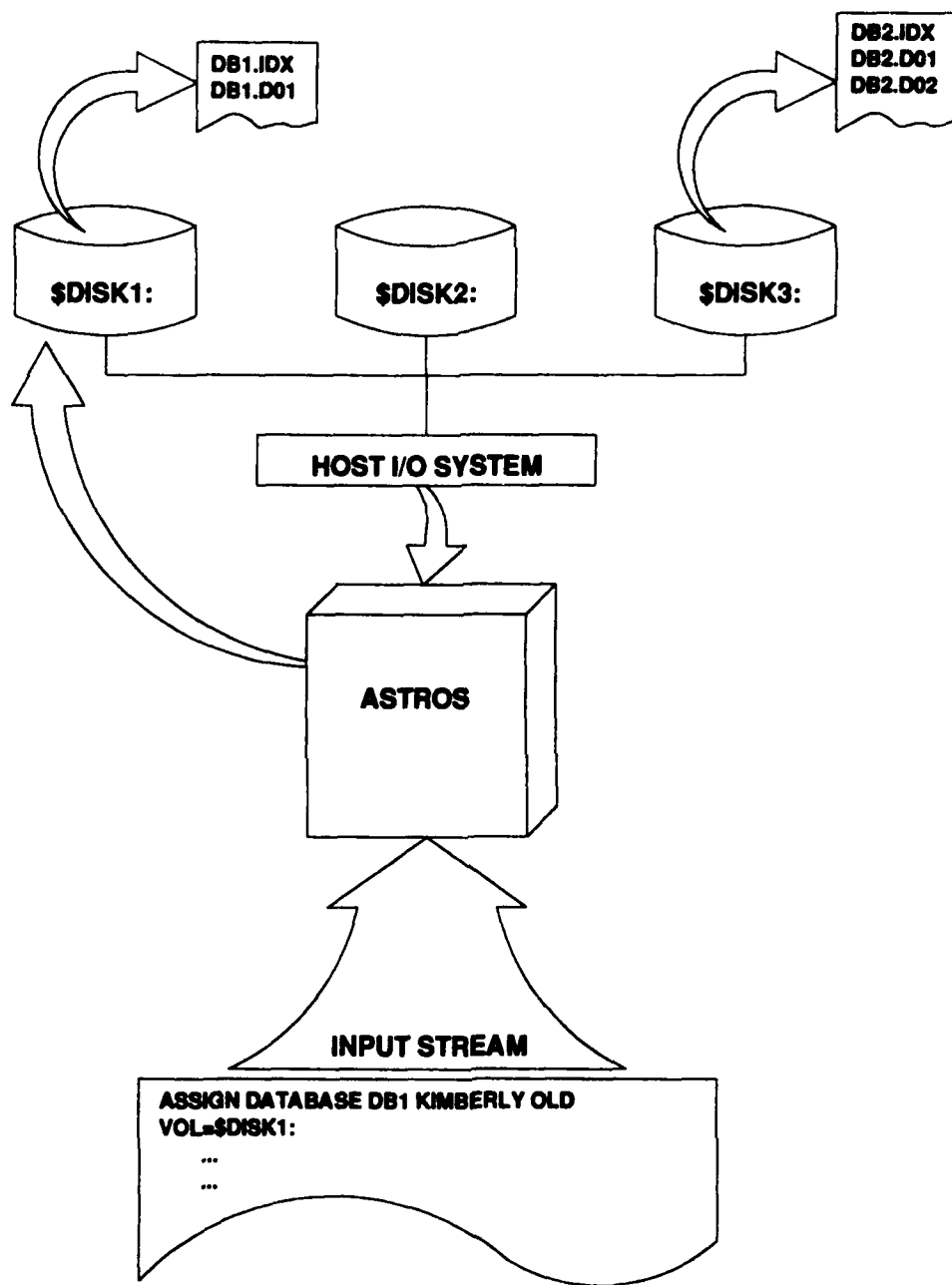


Figure 3. Function of the ASSIGN DATABASE Entry

1.3.1. ASSIGN DATABASE DESCRIPTIONS FOR HOST COMPUTERS

This section contains the descriptions of the machine and installation dependent parameters on the **ASSIGN DATABASE** entry for three machines on which ASTROS is currently functional. The parameters that are available at each site are listed and details of their use are presented. The user is cautioned that these are site dependent parameters which may be different for each installation even if the host system is the same. This documentation is provided both as an example to the system programmer and because the features that have been made available on these machines are very likely to exist on most machines that may be used. The user is referred to their ASTROS system manager for the particulars of the interface between the local host system and ASTROS.

The host systems documented in this section are:

- (1) DEC VAX computers using VMS
- (2) IBM 370/3090 computers using MVS
- (3) Computers using Unix-based Operating Systems

1.3.1.1. VMS IMPLEMENTATION

The **ASSIGN DATABASE** entry supplies the ASTROS system with the root name of the database files, the status of those files, and a set of user parameters. The root name is limited to 8 characters in length, the status is one of **NEW**, **OLD** or **TEMP** and the set of user parameters can be any of the following keyword commands:

VOL=name	Names the device and/or default directory on which the database files reside.
DBLKSIZE=n	CADDB data files block size in words.
IBLKSIZE=n	CADDB index file block size in words.
DELETE	Denoting that the run-time database files are to be deleted after the execution.
KEEP	Denoting that the run-time database files are to be returned after the execution.

DCL Requirements:

Under MicroVMS and VMS, the database files can be explicitly named in the FORTRAN open statement or an equivalence between a logical name and the true file name can be established using the DCL ASSIGN command:

```
$ ASSIGN <physical-name> <logical-name>
```

Both options can be used in the VMS version of ASTROS. The logical names **dbrootIDX** and **dbrootD01** are checked to see if they have been ASSIGNED. If so, these names are used in the Fortran OPEN and the ASSIGN provides the hook to the physical files. When no logical names exist, the filenames:

```
dbroot.idx  
dbroot.d01
```

are used. When a VOL parameter is specified, a file name with the structure of a physical file name is used:

```
voldbroot.IDX  
voldbroot.DO1
```

As an example, assuming that the logical names **MULTIDX** and **MULTDO1** do not exist:

```
ASSIGN DATABASE MULT SHAZAM NEW
```

would use **MULT.IDX** and **MULT.DO1** as file names in Fortran OPEN statements whereas:

```
ASSIGN DATABASE MULT SHAZAM NEW VOL = $DISK1:[SCR]
```

would use **\$DISK1:[SCR]MULT.IDX** and **\$DISK1:[SCR]MULT.DO1** as filenames in the OPEN statements. Note that if the logical name is assigned, VOL will be ignored.

TEMP Data Base Example:

When the status is **TEMP**, a temporary database is created. Internally, the file is not named in the OPEN statement and has a status of **SCRATCH**. Generally, this means that the files will be created in the user's default directory. The only legal optional parameters are **DBLKSIZ** and **IBLKSIZ**. In the example:

```
ASSIGN DATABASE TEST PASS TEMP
```

the database files will be created using the default block sizes and deleted upon termination of the execution.

NEW Data Base Example:

When the status is **NEW**, a new database is created. The filename will be formed from the root name and, if present, from the VOL option as discussed. Any of the user parameters may be used. As an example:

```
ASSIGN DATABASE TEST PASS NEW
```

will create either the files **TEST.IDX** and **TEST.DO1** or, if the DCL contains the proper assignments:

```
$ ASSIGN SYS$SCRATCH:TEST.IDX TESTIDX  
$ ASSIGN SYS$SCRATCH:TEST.DO1 TESTDO1
```

the files named in the DCL ASSIGN statements.

The **DELETE** option will cause the files to be deleted using the **DISP='DELETE'** option on the OPEN and CLOSE statements. Otherwise the files will be kept.

OLD Data Base Example:

When the status is **OLD**, an old database is used. The physical files that make up the database must exist. Only the VOL parameter is legal and the database files will be kept after the execution has terminated. In the example:

```
ASSIGN DATABASE TEST PASS OLD VOL = $DISK1:[DB]
```

the pre-existing database files

```
$DISK1:[DB]TEST.IDX  
$DISK1:[DB]TEST.DO1
```

will be used in the ASTROS execution.

1.3.1.2. IBM MVS IMPLEMENTATION

The **ASSIGN DATABASE** entry supplies the ASTROS system with the root name of the database files, the status of those files, and a set of user parameters. The root name is limited to 5 characters in length, the status is one of **NEW**, **OLD** or **TEMP** and the set of user parameters can be any of the following keyword commands:

DBLKSIZ=n CADDDB data files block size in words.

IBLKSIZ=n CADDDB index file block size in words.

NEW Data Base Example:

When the status is **NEW**, a new database is created. The **NEW** refers to the fact that the database itself is to be initialized. The physical files that make up the database may exist or be created with the proper DD card. Other legal parameters are **DBLKSIZ** and **IBLKSIZ**. Here is an example:

```
ASSIGN DATABASE RICH RICHPASS NEW
```

JCL should be as follows:

```
//RICHIDX DD DSN=RICHDB.INDEX.SPACE=(TRK,10),DISP=(,CATLG)  
//RICHDO1 DD DSN=RICHDB.DATA,SPACE=(TRK,50),DISP=(,CATLG)
```

In this example a database with files **RICHDB.INDEX** and **RICHDB.DATA** will be created.

JCL Requirements:

On the IBM computer the user must supply the appropriate JCL cards for each database file. The DDNAMEs for these cards are built from **dbroot** root file name as follows:

```
dbrootIDX - index file  
dbrootD01 - First data file  
dbrootD02 - Second data file
```

The only required DD parameters for new files are **SPACE** and **DISP**. The user can control the number of data files by including the desired number of DD statements.

TEMP Data Base Example:

When the status is **TEMP**, a temporary database is created. Internal to the database, there is no difference between a **TEMP** and a **NEW** database on the IBM. The only difference is in the JCL where the user supplies a temporary file instead of a permanent one. The only legal optional parameters are **DBLKSIZ** and **IBLKSIZ**. Here is an example:

```
ASSIGN DATABASE RICH RICHPASS TEMP DBLKSIZ=2048 IBLKSIZ=256
```

JCL should be as follows:

```
//RICHIDX DD DSN=&INDEX,SPACE=(TRK,10)
//RICHDO1 DD DSN=&DATA,,SPACE=(TRK,50)
```

OLD Data Base Example:

When the status is **OLD**, an old database is used. The physical files that make up the database must exist. No other parameters are legal. Here is an example:

```
ASSIGN DATABASE RICH RICHPASS OLD
```

The JCL should then be as follows:

```
//RICHIDX DD DSN=RICHDB.INDEX,DISP=OLD
//RICHDO1 DD DSN=RICHDB.DATA,DISP=OLD
```

1.3.1.3. UNIX SYSTEM IMPLEMENTATION

The **ASSIGN DATABASE** entry supplies the **ASTROS** system with the root name of the database files, the status of those files, and a set of user parameters. The root name is limited to 8 characters in length, the status is one of **NEW**, **OLD** or **TEMP** and the set of user parameters can be any of the following keyword commands:

VOL=name	Names the disk volume (directory) on which the database files reside. VOL and DIR are synonymous and mutually exclusive.
DIR=name	Names the disk volume (directory) on which the database files reside. VOL and DIR are synonymous and mutually exclusive.
DBLKSIZ=n	CADDB data files block size in words.
IBLKSIZ=n	CADDB index file block size in words.
DELETE	Denoting that the run-time database files are to be deleted after the execution.
KEEP	Denoting that the run-time database files are to be returned after the execution.

TEMP Data Base Example:

When the status is **TEMP** a temporary database is created and no data is kept after the run. The only legal optional parameters are **DBLKSIZ** and **IBLKSIZ**.

```
ASSIGN DATABASE RICHDB RICHPW TEMP DBLKSIZ=2048 IBLKSIZ=256
```

NEW Data Base Example:

When the status is **NEW**, a new database is created. If the files already exist, they will be overwritten.

```
voldbroot.IDX - index file  
voldbroot.D01 - data file 1
```

The **VOL** parameter and the root database name are used to form the names. Other legal parameters are **DBLKSIZ** and **IBLKSIZ**.

```
ASSIGN DATABASE RICHDB RICHPASS NEW DELETE, VOL=/tmp/
```

In this example, a database with files **/tmp/RICHDB.IDX** and **/tmp/RICHDB.D01** will be created. If existing files are found they will be overwritten.

OLD Data Base Example:

When the status is **OLD**, an old database is used. The physical files that make up the database must exist. Two files are used to store the database. The names of these files are as follows:

```
voldbroot.IDX - index file  
voldbroot.D01 - data file 1
```

The **VOL** parameters and the root database name are used to form the names. No other parameters are legal. Here is an example:

```
ASSIGN DATABASE RICHDB RICHPASS OLD VOL=/home/rich/
```

In this example a database with files **/home/rich/RICHDB.IDX** and **/home/rich/RICHDB.D01** will be used.

1.4. THE INCLUDE DIRECTIVE

The input data stream typically resides in a single file, but the user can direct the input stream interpreter to include other files through the use of the **INCLUDE** directive in the primary input stream. The format of the **INCLUDE** command is:

INCLUDE <filename>

where,

filename is a name identifying the file to be included (maximum of 72 characters).

The filename, which is used in a FORTRAN **OPEN** statement, must satisfy the requirements of the particular host system for file names. Beyond this restriction, the user is free to have any set of contiguous non-blank characters in the filename. In order to avoid the possibility of an infinite recursion, there is a restriction on the include feature that no **INCLUDE** statement can appear in a file that is being included. For example, if the file "TENBAR" is being included, it may not itself contain an **INCLUDE** directive. The input stream interpreter will terminate with an appropriate error message should this occur.

The **INCLUDE** directive can appear anywhere in the input stream after the **ASSIGN DATABASE** entry. The **ASSIGN DATABASE** must always appear as the first non-blank line in order to allow the use of the run time database in the subsequent input stream interpretation. A single data packet can be split among included files or an **INCLUDE** file may contain parts of multiple data packets. The input interpreter merely replaces the **INCLUDE** directive with the data contained in the named file so the only requirement is that the input stream that results from the combination of all **INCLUDEs** have the form of a normal input stream. The **INCLUDE** feature can be very useful in certain circumstances. For example, a special user developed MAPOL sequence can be stored and maintained external to the files containing the engineering data for particular runs; or, conversely, the bulk data representing a large model can be included into the file containing the solution control directives.

1.5. THE DEBUG PACKET

The debug packet represents a development tool and is intended to be used primarily by those responsible for maintaining the software. The debug packet provides the system programmer with the means to invoke or control certain executive and database management system functions that are helpful in tracking the ASTROS execution and/or testing the executive and database management system software. However, because some of the debug options can be useful to the general user, the debug packet is fully documented in the User's Manual rather than in the Programmer's Manual. This section documents each of the debug options and indicates how the option can be useful in debugging the ASTROS procedure. Emphasis is placed on those debug options that are of interest to the general user.

The debug packet is initiated by the keyword **DEBUG**, which must appear alone on the line of the input stream that follows the **ASSIGN DATABASE** entry and that precedes any other data packet. Following the initiator, any number of debug lines can be included in the data stream. Each debug command line can be composed of a number of debug commands, appearing in any order, separated by blanks or commas. The **DEBUG** packet is terminated when a new data packet initiator, or the end of the input stream, is encountered. Most debug commands consist of single keywords which toggle flags activating the debug functions. The appearance of these debug keywords is all that is required to activate the option. Other debug commands select that a flag take on a particular value. These commands have the form:

`<command> = <value>`

There can be any number of blanks between the end of the command keyword, the value and the equal sign, but neither the command nor the value can contain imbedded blanks. Any errors in the **DEBUG** packet input will result in warnings but will not terminate the execution and the erroneous command will be ignored.

The tables shown in this section the list of keywords that can be included in the **DEBUG** packet. The debug commands are grouped into executive system and database management system debugs. Each of these groups is described in greater detail in the following sections.

1.5.1. EXECUTIVE SYSTEM DEBUG COMMANDS

The first four executive system keywords are intended to assist the system programmer in following the actions of the MAPOL compiler and execution monitor. The options are shown in Table 2.

As such, they are of limited value to the general user. The **MATRIX** option, however, can be useful in tracking the execution of the MAPOL program. It echoes the matrix utility calls for all matrix operations that are in the MAPOL sequence. For example, if the MAPOL program includes the expression:

```
[A] := TRANS( [B] ) * [C] + [D];
```

the **MATRIX** trace echoes the resultant call to the **MPYAD** large matrix utility with the arguments shown in detail. This trace can be very useful in determining which particular MAPOL instruction is being executed when a problem occurs. Large MAPOL programs with many loops and a large number of matrix expressions can be debugged quite simply using the **MATRIX** trace. All MAPOL statements that result in calls to any of the large matrix utilities, such as **PARTN**, **MERGE**, **MPYAD**, and **MXADD** are echoed.

LOGBEGIN and **LOGMODULE** provide expanded echoes of the module timing summary that is found at the end of each ASTROS output file. When problems cause early termination of the job, these options provide the name of the last module entered prior to the failure. This provides a starting point to diagnose the problem.

Table 2. MAPOL Debug Commands

KEYWORD	DESCRIPTION
MSTACK	MAPOL compiler stack output
MEXEC	MAPOL execution debug flag
MOBJ	MAPOL object code debug packet
MTRACE	MAPOL trace debug output
MATRIX	MAPOL peeper matrix operation trace
LOGMODULE	Expanded log entries for each module
LOGBEGIN	Beginning entries for each module in log file

1.5.2. DATABASE AND MEMORY MANAGER DEBUG COMMANDS

The database management system has a number of debug options which can be divided into three categories: trace options, control options and memory manager options. These are shown in Table 3.

The first group of database debug commands contain two tracing options: **TRACE** and **IOSTAT**. The **IOSTAT** keyword selects either a **FULL** tracing or a **SUMMARY**. The first of these options and the **IOSTAT=FULL** option are further controlled by the **ENTITY** option which completes the first group of keywords. Note: the tracing keywords generate an overwhelming amount of data which are often of limited use unless the user is familiar with the internal structure of the database files. The **ENTITY** keyword limits the activation of the tracing options to those times when the named database matrix, relation or unstructured entity is open. If no entity specification is made, the traces are active for all database operations. In addition to their role in debugging the database software, the trace options provide a useful means of debugging the interface between a user written module and the database.

The database control options **CALLSTAT** and **NOCOREDİR** provide user control over two internal database functions. The **CALLSTAT** option compiles a summary of the number of calls made to each database subroutine. This summary, in combination with the **IOSTAT** option, provides statistics on the number of database operations in the execution. The **NOCOREDİR** option is made available for machines with limited core memory resources. If **NOCOREDİR** is selected, the database manager stores the database directories on the database files rather than in core. This can substantially reduce the database memory requirements at the cost of increasing the number of input/output operations.

Table 3. Database Debug Commands

KEYWORD	DESCRIPTION	
TRACE	Traces all database CALLs	
IOSTAT=parm	Database I/O tracing	
	FULL	Full trace of I/O activity
	SUM	Summary of I/O activity
ENTITY=name	Restricts tracing to entity name	
CALLSTAT	Compiles statistics on the number of calls to each database routine at the end of a job.	
NOCOREDİR	Turns off the option to store directories in core	
NODELAYCRE	Turns off the option that delays entity creation until the entity is opened	
MEMORY	Memory manager debug print	

The last group of database debug options consists of the **MEMORY** command. This option causes an echo of all the memory management calls made in the modules. The user can then track the ASTROS execution into the engineering modules themselves. In addition to the echo, the **MEMORY** option invokes a *checksum* operation which checks for the integrity of the memory block headers on every memory manager operation. If the checksum fails, a message is written to the effect that a block header has been overwritten. This option is very effective in uncovering errors in engineering modules that make use of dynamic memory allocation.

1.5.3. INTERMEDIATE RESULTS PRINTING COMMANDS

Many of the ASTROS engineering modules have intermediate output print options that are useful in tracing the details of an analysis or in reviewing the quality of the inputs. These many options are listed in Table 4.

Table 4. Intermediate Results Debug Commands

KEYWORD	DESCRIPTION	
STEADY=n	Steady preface USSAERO output	
	1	Prints steady aerodynamic model geometry
	2	Includes stability coefficient data
	3	Includes pressure data
	4	Includes velocity components and matrix output
STEADYNP=n	Nonplanar preface USSAERO output	
	1	Prints steady aerodynamic model geometry
	2	Includes stability coefficient data
	3	Includes pressure data
	4	Includes velocity components and matrix output
AMP=n	Intermediate unsteady Aero matrices	
	1	Prints the SKJ matrix and, if only one group, includes AJJ, QKJ and QJJ if they exist
	>1	Includes D1JK, D2JK and AJJT matrices
FLUTTRAN=n	Additional flutter eigenextraction information	
	1	Prints the number of iterations required to find each flutter root
	>1	Includes the estimated roots for each iteration

Table 4. Intermediate Results Debug Commands - Continued

KEYWORD	DESCRIPTION	
SCEVAL=n	Prints additional constraint data	
	>0	Prints the stress or strain components, the constraint type and the constraint value for each constrained element/layer.
DESIGN=n	1	Prints initial design information
	2	Includes function values at each iteration
	3	Includes internal Microdot parameters
	4	Includes search directions
	5	Includes gradient information
	6	Includes scaling information
	7	Includes one-dimensional search information
MKUSET=n	MKUSET redundant set warnings	
	>0	Prints warning messages if the same degree of freedom is placed in a set more than once.
VANGO=n	VANGO algorithm output	
	1	Prints all output not controlled by OCPARM bulk data entry.
OCAPPROX=n	VANGO use of approximate problem	
	>0	VANGO iterates to convergence on the approximate problem rather than making one design variable change and then requiring a complete reanalysis.
OCINVERSE=n	VANGO use of inverse design variables	
	>0	VANGO uses inverse design variables when there are no shape function global variables. Using this with the OCAPPROX debug makes VANGO behave exactly like DESIGN in its treatment of the approximate problem. The only difference is that OC methods are used to update {v}.
SAROEOM=n	Planar steady aerodynamics geometry option	
	>0	STEADY stops the execution of ASTROS after the steady aerodynamic geometry has been computed. No printed output is generated unless the STEADY debug is also used.

Table 4. Intermediate Results Debug Commands - Continued

KEYWORD	DESCRIPTION	
NPSGEOM=n	Nonplanar steady aerodynamics geometry option	
	>0	STDYNP stops the execution of ASTROS after the nonplanar steady aerodynamic geometry has been computed. No printed output is generated unless the STEADYNP debug is also used.
MPSCAL=opt	Controls scaling of design variables	
	ON	Scales global variables to unity before Microdot is invoked (Default).
	OFF	Does not scale variables.
PLYDATA=opt	PRINT	Selects the output of PSHELL/MAT2 Bulk Data entries for elements defined by PCOMP1 data.
	PUNCH	
	BOTH	

1.5.4. SEQUENCER INTERMEDIATE PRINT COMMANDS

There are a number of print and control options for the grid point sequencer that are shown in Table 5.

Table 5. Sequencer Debug Commands

KEYWORD	DESCRIPTION	
SEQPRINT=opt	Selects sequencing intermediate print	
	DETAIL	Requests detailed print of sequencing
	DIAGNOSTIC	Requests diagnostic print
SEQMETH=meth	Select sequencer method	
	CM	Selects the Cuthill-McKee Method
	GPS	Selects the Gibbs-Poole-Stockmayer Method
	ALL	Selects the best of both 1 and 2
NOSEQMPC	Requests that MPCs not be processed during sequencing	
SEQCRIT=crit	Selects sequencing method	
	BAND	Selects minimum bandwidth criteria
	PROF	Selects profile criteria
	RMS	Selects RMS criteria
	WAVE	Selects minimum wavefront criteria
SEQPUNCH	Requests punching of SEQGP bulk data entries	
SEQOFF	Deselects sequencing	

2. THE EXECUTIVE SYSTEM AND MAPOL

The ASTROS system is an executive controlled software procedure. One of the functions of the ASTROS executive system, described in detail in Reference 1, is to determine the sequence in which the modules of the procedure are invoked. For ASTROS, the Matrix Analysis Problem Oriented Language (MAPOL) has been developed to perform this executive system task. The MAPOL language has its conceptual roots in the Direct Matrix Abstraction Program (DMAP) capability developed for the NASTRAN structural analysis system (Reference 2). The DMAP "language," used to create NASTRAN's solution algorithms is very crude; however, it has been a major factor in extending the life cycle of the software. It provided a simple method of installing new code and functional capabilities into the system and afforded the user an opportunity to interact with the software. MAPOL provides the same advantages to the ASTROS system and represents a considerable advance over DMAP in that MAPOL is a structured, procedural language that directly supports high order matrix operations, manipulation of data base entities and complex data types. Moreover, the syntax of the language looks much like that of any scientific programming language and so is easily learned by anyone who knows FORTRAN or PASCAL.

From the user's point of view, ASTROS is directed by a sequence of control statements "coded" in the MAPOL language just as a NASTRAN rigid format is coded in the DMAP "language." **As with the NASTRAN Rigid Formats, the majority of users will use the standard MAPOL sequence. This is the default, and as such it requires no special action. Advanced users may optionally edit the standard sequence or write their own "program". The methods used to do this are described in this Chapter. Because changes to the executive system are an advanced topic, first-time users may proceed directly to Chapter 3.**

The executive system within ASTROS compiles the MAPOL program and executes the resultant "ASTROS machine code" which directs the execution of the ASTROS procedure. (Note that ASTROS is NOT written in MAPOL, only the executive control algorithm is written in the MAPOL language. In fact, ANSI standard FORTRAN was used to write the compiler for MAPOL and for all the engineering software of the ASTROS system.) MAPOL allows the user to manipulate the software system in many ways to tailor the available capabilities to perform particular tasks. At a higher level of sophistication, the user may add modules to the system or replace modules that already exist. Obviously, some of these features require a knowledge of the ASTROS system that is beyond the scope of the User's Manual. Those features that require detailed information are more fully discussed in the Programmer's Manual but their existence is emphasized here in order to introduce the user to the flexibility that the executive system provides.

This section serves two functions. First, it presents the mechanics of the MAPOL packet and, second, it presents the standard MAPOL sequence that has been developed to direct the optimization and analysis tasks for which ASTROS has been designed. The potential of the executive system to tailor the

ASTROS procedure will be explored in this discussion of the standard sequence. In addition to this Chapter, Chapter 6 presents a detailed description of the MAPOL language, its syntax and features. Included in this chapter is a listing of the most recent ASTROS standard executive sequence. It cannot be overemphasized that, while the capabilities implemented in the ASTROS software are significant, the true power embodied in the ASTROS system is its immense flexibility, largely provided by the executive system and its MAPOL language.

The MAPOL packet is initiated either by the keyword MAPOL or by the keyword EDIT and is terminated upon encountering the SOLUTION CONTROL packet, the BULK DATA packet or the end of the input stream. In addition, each of the initiator keyword commands act as directives to the MAPOL compiler to take specific actions. The exact form of the MAPOL and EDIT commands is:

$$\text{MAPOL} \begin{bmatrix} \text{GO} \\ \text{NOGO} \end{bmatrix} \begin{bmatrix} \text{LIST} \\ \text{NOLIST} \end{bmatrix}$$

$$\text{EDIT} \begin{bmatrix} \text{GO} \\ \text{NOGO} \end{bmatrix} \begin{bmatrix} \text{LIST} \\ \text{NOLIST} \end{bmatrix}$$

where:

$\begin{bmatrix} \text{GO} \\ \text{NOGO} \end{bmatrix}$ selects whether the MAPOL program is to be executed after compilation.

$\begin{bmatrix} \text{LIST} \\ \text{NOLIST} \end{bmatrix}$ selects whether the MAPOL source code is to be written to the output file.

The MAPOL command is followed by a MAPOL program which can be any syntactically complete set of MAPOL statements as described in the chapter on MAPOL Programming (Chapter 6). The EDIT command indicates that the MAPOL packet will consist of edit commands that INSERT, DELETE or REPLACE lines of the standard executive sequence.

2.1. THE MAPOL PROGRAM

If the MAPOL packet begins with the MAPOL command line, the compiler assumes that the remaining statements in the packet constitute a complete MAPOL program. That program can be any set of MAPOL statements that satisfy the rules of the language as presented in Chapter 6. The program can call any of a number of intrinsic functions (including most of the common FORTRAN intrinsic functions) and any of the "engineering" utilities and modules that have been defined to the compiler. The user can access these modules in any desired order, subject only to limits imposed by the engineering modules themselves. In addition, the user can write special purpose modules and define them to the compiler through the SYSTEM GENERATION (SYSGEN) program discussed in the Programmer's Manual. Thus, a wide range of tasks can be performed using the ASTROS system in combination with a user's MAPOL program.

The MAPOL language can be read and written easily by anyone familiar with a scientific programming language. This feature opens the advantages of the executive system to the average user without requiring specialized knowledge in computer science or requiring effort to learn a radically

different programming language. The user will often find the simplicity and power of the MAPOL language enables many tasks to be performed using the ASTROS system that are not explicitly supported in the standard executive sequence.

2.2. MAPOL EDIT COMMANDS

If the MAPOL packet begins with the EDIT command line, the compiler assumes that the remainder of the packet (if any) is composed of MAPOL edit commands and new MAPOL statements that modify the standard executive sequence. The set of edit commands is given in Table 6 (and in Chapter 6). They allow the user to insert, delete and replace lines of the standard MAPOL sequence. All the edit commands reference a line number or range of line numbers. The line numbers are those in a compiled listing of the standard MAPOL sequence which is written as part of the system generation task. When editing the standard sequence, the user is cautioned to obtain the most recent listing either from the SYSGEN output or by executing ASTROS with an input stream containing only an ASSIGN DATABASE entry and the one line MAPOL packet:

EDIT LIST NOGO

This input stream will result in an output file containing the current listing of the standard executive sequence.

Table 6. MAPOL Edit Commands

STATEMENT	FUNCTION
EDIT	Modify the standard solution
DELETE a[,b]	Remove lines a through b inclusive
REPLACE a[,b]	Removes lines a through b inclusive and replaces them with the following lines
INSERT a	Insert the lines following the command after line a

2.3. THE STANDARD EXECUTIVE SEQUENCE

As previously mentioned, the MAPOL language has its conceptual roots in the DMAP "language". In order to allow the user of NASTRAN to perform certain predefined analyses, a set of "rigid formats" or DMAP algorithms were written, alleviating the user of the need to learn the details of the control language. Each rigid format allowed the user to perform analyses in a different engineering discipline; for example, static structural analyses, normal modes analyses, or transient analyses. In a similar manner, a standard executive sequence or MAPOL algorithm has been developed for the ASTROS system which supports all the engineering disciplines and optimization features of the procedure. Unlike the multiple DMAP rigid formats, however, there is a single MAPOL sequence that supports all the available engineering disciplines as well as optimization. This fundamental difference is necessary to permit multidisciplinary optimization.

One consequence of having a single multidisciplinary algorithm is that the standard sequence appears to be very complicated. The purpose of this subsection is to present the internal structure and flow of the standard MAPOL sequence thereby providing the user with sufficient information to tailor the standard sequence to suit individual needs. The discussion in this section will be general in order to provide the necessary overview and to introduce the concepts embodied in the standard sequence. Modifications to the standard sequence will be presented primarily in terms of capabilities but the presentation will be supported by examples that represent both simple and more complex modifications. Finally, the Chapter closes with a detailed line-by-line presentation of the standard executive sequence. The reader is also referred to the Programmer's Manual for information on the addition of modules to the ASTROS engineering library.

2.4. STANDARD EXECUTIVE SEQUENCE STRUCTURE

The standard MAPOL sequence consists of two major components: the variable declarations and the solution algorithm. The solution algorithm can be further divided into preface modules, the optimization segment and the final analysis segment. The declaration segment declares all variables used in the MAPOL sequence. This includes all integer and real scalar variables as well as high order variables: relations, matrices and unstructured data base entities. Within the solution algorithm, the preface modules comprise a group of engineering modules exercised prior to the boundary condition loops to perform a number of system initialization tasks; e.g. loads generation and the computation of invariant aerodynamic matrices. The separate optimization and analysis segments consist of a loop on the number of (optimization or analysis) boundary conditions in the current execution. In the optimization segment, a second boundary condition loop is performed to obtain the sensitivities of active boundary condition dependent constraints in preparation for the optimization task.

Figure 4 provides the standard algorithm structure showing how multidisciplinary optimization is performed in ASTROS. It is readily apparent that the structure of the standard MAPOL sequence has been determined by the requirement to perform multidisciplinary optimization. Each of the segments of the standard sequence are discussed in greater detail in the following subsections.

PREFACE SEGMENT		Initialization (PREFACE) Segment
		WHILE NOT CONVERGED DO
OPTIMIZATION SEGMENT	ANALYSIS PHASE	For Each Boundary Condition Do Discipline 1 Subcase 1 Constraints Subcase 2 Constraints ... Discipline 2 ... End Do
	SENSITIVITY PHASE	Select Active Constraints For Each Active Boundary Condition Do Active Discipline 1 Active Subcase 1 Constraint Sensitivities Active Subcase 1 Constraint Sensitivities ... Active Discipline 2 ... End Do
	OPTIMIZATION PHASE	Redesign Based on Current Active Constraints and Constraint Sensitivities
		END DO
FINAL ANALYSIS SEGMENT		For Each Boundary Condition Do Discipline 1 Subcase 1 Subcase 2 ... Discipline 2 ... End Do

Figure 4. Structure of the Standard MAPOL Sequence

2.4.1. MAPOL Declarations

MAPOL is a strongly typed language that requires all variables used in a program unit (either the main program or a procedure) to be declared. This applies to both simple variables like real and integer scalar or array variables and to high order variables (like MATRIX) that refer to data base entities. The first several hundred lines of the standard sequence consist solely of these variable declarations. Tables 7 through 12 give a summary of the scalar parameters used in the standard MAPOL sequence. These parameters, set in engineering modules or in the MAPOL sequence, are used as logic control flags and/or arguments to the engineering modules. The tables, which are separated by function, provide a brief description of each variable and a list of modules (where applicable) that use the parameter. For a description of all the variables used as arguments of the engineering modules, the reader is referred to the ASTROS Programmer's Manual. It should be noted that all of these variables can be directly modified within the MAPOL algorithm at the user's discretion. A discussion of those parameters that the user is most likely to want to modify is given in Section 2.4.3, but the experienced user is free to change any variable in the MAPOL sequence.

Higher order variables fall into two categories: MAPOL entities and *hidden* entities. MAPOL entities are those that actually appear in the MAPOL sequence while *hidden* entities are those that are declared but do not subsequently appear in the sequence. Their declaration ensures that the corresponding data base entity is created and can be used by a number of engineering modules without requiring the entity name to appear in the argument list. *Hidden* entities are typically those that contain the raw data needed by many modules; e.g. bulk data, geometry data and connectivity data. The declarations of the higher order variables are arranged to place logically related entities together. Several of the matrix entities, it should be noted, are subscripted, for example [KLLINV(1000)]. The subscripted matrix entity allows the ASTROS procedure to perform multiple analyses in several boundary conditions and retain the information needed to compute the sensitivities of the active constraints retained from each of these boundary conditions. The ASTROS executive system generates a name for each subscripted variable and that name is used by all the engineering modules receiving the subscripted entity name as an argument. The actual data base entity name need not be known. This does, however, impose the following restriction: a subscripted entity may not be used as a *hidden* entity in any engineering module; it must appear in the calling list for the module because only the executive system knows the actual name of the data base entity corresponding to the current subscript value. In the standard sequence, provision has been made for up to 1000 entities (doubly subscripted arrays of entities are set up for 30 boundary conditions and 33 secondary subscript values), but the user can change the declared number of subscripts to match the required range of indices.

Table 7. Real Parameters in the Standard Sequence

PARAMETER NAME	USED IN MODULES	DESCRIPTION	DEFAULT
ALPHA	SOLUTION FSD	Exponential move limit for fully stressed design. Set through the Solution Control OPTIMIZE command.	0.90
BQDP	BLASTFIT	Dynamic pressure value used in the current nuclear blast subcase. Output from BLASTFIT and used subsequently in MAPOL expressions.	
CNVRGLIM	DESIGN FSD VANGO	Convergence test limit specifying the maximum percent objective change for the appropriate problem to be considered converged. Output from SOLUTION.	0.50
CTL	ACTCON DESIGN FSD, VANGO	Criteria for defining a constraint to be active in determining convergence in ACTCON. If value > CTL, the constraint is active. Set in DESIGN, VANGO or FSD.	
CTLMIN	ACTCON DESIGN FSD, VANGO	Criteria for denoting a constraint to be feasible in determining convergence in ACTCON. If maximum constraint value < CTLMIN, the design is feasible. Set in DESIGN, VANGO, or FSD.	
EPS	SOLUTION ACTCON	Criteria used in ACTCON for selecting active constraints. All constraints with values greater than EPS will be retained. Set through the Solution Control OPTIMIZE command. (See also NRPAC)	-0.10
FMAX	GRD1 GRD2	The maximum frequency value associated with the NRV eigenvalues computed for dynamic reduction in the current boundary condition. Output from QDR1.	
MACH	SAERODRV	Mach number for the current case. Set in SAERODRV.	
MOVLIM	SOLUTION DESIGN MAKDFV TCEVAL	A move limit applied to the physical design variable (v) for mathematical programming methods. The move is: $v/\text{MOVLIM} < v < v + \text{MOVLIM}$. Set through the Solution Control OPTIMIZE command.	2.0 (≥1.0)
NRPAC	SOLUTION ACTCON	Criteria used in ACTCON for selecting active constraints. At least NRPAC times NDV (see Table 5) constraints will be retained. Set through the Solution Control OPTIMIZE command. (See also EPS)	3.0
OCMOVLIM	SOLUTION VANGO	A move limit applied to the physical design variable (v) for the optimality criteria method. See MOVLIM.	10^{-5}
QDP	SAERODRV SAERO others	Dynamic pressure value used in the current steady aeroelastic subcase. Output from SAERODRV used subsequently in MAPOL expressions and modules.	
WINDOW	TCEVAL	<p>The window in which the MOVLIM bound is overridden to allow local variables to change sign. If WINDOW is 0.0, then the local variable may not change sign. If it is nonzero, the half-width of a band around zero, called EPS, is computed by:</p> $\text{EPS} = \text{WINDOW}/100 * \text{MAX}(\text{ABS}(\text{TMIN}), \text{ABS}(\text{TMAX}))$ <p>If the local variable falls within the band, then the new minimum or maximum for the current iteration is changed to lie on the other side of zero from the local variable. Output from SOLUTION.</p>	0.0 (≥0.0)

Table 8. Integer Modelling Parameters

NAME	MODULES	DESCRIPTION
ASIZE	GDR3	The number of a-set degrees of freedom.
BC	N/A	Boundary condition loop counter.
DDFLG	DDLOAD	Indicates if the current static subcases contain design dependent (gravity or thermal) loads. Output by DDLOAD.
ESIZE	BCBGPDT others	The number of extra points in the boundary condition.
GNORM	GDR3	The sum of LJSET and LKSET.
GSIJZB	IFP others	The number of structural degrees of freedom in the model. Output from IFP and subsequently used in many modules.
GSIJZ	GDR4 others	GSIJZB modified subject to dynamic reduction.
HSIJZ	FLUTTRAN OFFPEDR REIG others	Number of eigenvectors extracted by the REIG module. Set in REIG.
LJSET	GDR1	Number of degrees of freedom in the j-set in dynamic reduction. Set in GDR1.
LKSET	GDR1	Number of degrees of freedom in the j-set in dynamic reduction. Set in GDR1.
LSIJZ	GDR1	The number of l-set degrees of freedom.
NEIV	GDR1 GDR2	An output from GDR1 indicating the number of eigenvalues below the maximum frequency specified for dynamic reduction.
NGDR	BOUND	Logical flag equal to negative one if dynamic reduction is selected for the current boundary condition.
NMPC	BOUND ABOUND	Logical flag equal to the number of degrees of freedom in the multipoint constraint set for the current boundary condition.
NOMIT	BOUND ABOUND	Logical flag equal to the number of omitted degrees of freedom in the current boundary condition.
NRSET	BOUND ABOUND	Logical flag equal to the number of support degrees of freedom in the current boundary condition.
NSPC	BOUND ABOUND	Logical flag equal to the number of single point constraint degrees of freedom in the current boundary condition.
PSIJZ	BCBULK others	The number of grid and extra point degrees of freedom.
SINGASET SINGLSET SINGOSET	SDCOMP	Used to process singular matrix columns.

Table 9. Integer Design Parameters

NAME	MODULES	DESCRIPTION
FSDE	SOLUTION	The last iteration to use fully stressed design. Output from SOLUTION.
FSDS	SOLUTION FSD	The first iteration to use fully stressed design. Output from SOLUTION.
MAXITER	SOLUTION ACTCON	Parameter set in the MAPOL sequence indicating the maximum number of resizing cycles that are to be performed. Set through the Solution Control OPTIMIZE command. (Def = 15)
MPE	SOLUTION	The last iteration to use mathematical programming. Output from SOLUTION.
MPS	SOLUTION FSD	The first iteration to use mathematical programming. Output from SOLUTION.
NACSD	ABOUND	The number of active stress and displacement constraints in the current active boundary condition. Used to select either the virtual load or gradient method in sensitivity analysis. Set in ABOUND.
NAUS	ABOUND	The number of active displacement vectors for statics. Set in ABOUND.
NBNDCOND	SOLUTION	The total number of boundary conditions in the solution control packet. Equal to the number of optimization boundary conditions plus the number of analysis boundary conditions. Output from SOLUTION.
NDV	MAKEST, others	The number of global design variables in the design model. Set by MAKEST and used in many subsequent modules.
NITER	N/A	The current optimization iteration number.
NUMOPTBC	SOLUTION	The number of optimization boundary conditions in the solution control packer. Set in SOLUTION.
OCE	SOLUTION	The last iteration to use optimality criteria. Set in SOLUTION.
OCS	SOLUTION FSD	The first iteration to use optimality criteria. Set in SOLUTION.

Table 10. Integer Aerodynamic Parameters

NAME	MODULES	DESCRIPTION
MINDEX	ABOUND, AEROSENS, BOUND, PFAERO	The index value for the Mach number dependent subscripted steady aerodynamic matrices. Typically has a value used to select the proper matrices for the current boundary condition.
SUB S	SAERODRV SAERO others	Identifies the subcase subscript. SAERO subcases with the same symmetry Mach number, MINDEX, trim type, and dynamic pressure are processed using the same subscript. This occurs with multiple load conditions with the same aero correction.
SYM	BOUND	A control flag denoting whether the symmetric (SYM=1) or antisymmetric (SYM=-1) steady aeroelastic matrices are to be used are to be used in the current boundary condition.

Table 11. Integer Discipline Parameters

NAME	MODULES	DESCRIPTION
BBLAST	BOUND	Indicates if there are any nuclear blast subcases in the current boundary condition.
BDFR	BOUND	Indicates if there are any direct frequency response subcases in the current boundary condition.
BDRSP	BOUND	Indicates if there are either transient or frequency response disciplines in the current boundary condition.
BDTR	BOUND	Indicates if there are any direct transient response subcases in the current boundary condition.
BDYN	BOUND	Indicates if there are any dynamic analyses (flutter, transient or frequency) in the current boundary condition.
BFLUTR	BOUND	Indicates if there are any flutter analyses in the current boundary condition.
BGUST	BOUND	Indicates if there are any gust loads for either transient or frequency disciplines in the current boundary condition.
BLOAD	BOUND	Indicates if there are any mechanical, thermal or gravity static applied loads in the current boundary condition.
BMASS	BOUND	Indicates if a mass matrix exists in the current boundary condition.
BMFR	BOUND	Indicates if there are any modal frequency response subcases in the current boundary condition.
BMODES	BOUND	Indicates if there are any disciplines that require that a normal modes analysis be performed.
BMTR	BOUND	Indicates if there are any modal transient response subcases in the current boundary condition.
BSAERO	BOUND	Indicates if there are any steady aeroelastic subcases in the current boundary condition.
DMODES	BOUND	Indicates if there are any modal disciplines in the current boundary condition.
NGDR	BOUND	Indicates if dynamic reduction is selected for the current boundary condition.

Table 12. Logical Discipline Parameters

NAME	MODULES	DESCRIPTION
ACTAEFF	ABOUND	TRUE if the current boundary condition has any active aeroelastic effectiveness constraints.
ACTAERO	ABOUND	TRUE if the current boundary condition has any active constraints associated with SAERO analyses.
ACTBOUND	ABOUND	TRUE if the current boundary condition has any active constraints.
ACTDYN	ABOUND	TRUE if the current boundary condition has any active frequency constraints.
ACTFLUT	ABOUND	TRUE if the current boundary condition has any active flutter constraints.
AEFLG	SAERO	Logical array which indicates whether the current SAERO subscript value has aeroelastic effectiveness constraints applied to it.
APPCNVRG	DESIGN ACTCON	TRUE when the approximate problem was converged in a previous iteration.
GLECNVRG	ACTCON	TRUE when global convergence has been reached.
K2GGFLG	MK2GG	Set TRUE in MK2GG if a K2GG matrix is input for the current boundary condition.
LOOP	-	General logical used to control DO-WHILE loops.
M2GGFLG	MK2GG	Set TRUE in MK2GG if an M2GG matrix is input for the current boundary condition.
NONONLY	NPSAERO	TRUE if the run contains any NPSAERO analyses.
PFLAG	ACTCON DESPUNCH	Set TRUE in ACTCON if DESPUNCH needs to punch a new model.

2.4.2. The Solution Algorithm

Finite element structural analysis, which forms the core of the ASTROS system, requires the manipulation of large matrices. The MAPOL control language was designed with this requirement in mind and, therefore, is able to directly support the manipulation of matrices. Consequently, the majority of the MAPOL sequence consists of matrix equations. The algorithmic nature of the MAPOL syntax allows the reader to follow these matrix operations fairly easily, and the notation roughly follows that used in the Theoretical Manual. Therefore, the focus of this section will be the description of modules called by the MAPOL sequence.

There are a number of engineering and utility modules called to perform tasks associated with the several analysis disciplines supported by the ASTROS system. Table 13 of Section 2.4.2.1 lists the modules defined to the ASTROS executive system, and provides a brief description of each. Not all of these modules appear in the standard solution sequence, but are included in the table to ensure its completeness and usefulness in modifying the standard sequence. The use of these modules is discussed in more detail in the section on modifying the standard MAPOL sequence and are more fully documented in the Programmer's Manual. The brief descriptions of the remaining segments of the standard algorithm that follow, coupled with the inherent readability of MAPOL syntax, are felt to provide a reasonably complete picture of the flow through the standard sequence.

2.4.2.1. MAPOL Engineering and Utility Modules

This section contains a brief description, shown in Table 13, of each of the MAPOL addressable modules defined to the ASTROS executive system. The intrinsic mathematical functions of the MAPOL language are not included.

Table 13. Summary of ASTROS Modules

MODULE	TYPE	DESCRIPTION
ABOUND	ENGINEERING	Generates flags for the current boundary condition during the sensitivity calculation. These are then returned to the executive sequence to direct the execution of the required sensitivity analyses.
ACTCON	ENGINEERING	Determines whether the design task has converged. If the optimization has not converged, this module selects which constraints are to be included in the current redesign. On termination or print request, this routine computes the values of the local design variables.
AEROEFS	ENGINEERING	Evaluates aeroelastic effectiveness sensitivities.
AEROSNS	ENGINEERING	Computes the sensitivities to active strength constraints and/or aeroelastic effectiveness constraints for active steady aeroelastic optimization boundary conditions.
AMP	ENGINEERING	Computes the discipline dependent unsteady aerodynamic matrices for flutter, gust and blast analyses.
ANALINIT	ENGINEERING	Initializes the final analysis pass.
APPEND	MATRIX	Appends one matrix to another.
AROSNSDR	ENGINEERING	Driver for SAERO sensitivity analysis.
AROSNSMR	ENGINEERING	Merges SAERO sensitivities for each subscript into [MATOUT] in case order for active subcases.
BCBGPDT	ENGINEERING	Builds the boundary condition dependent grid point coordinate relation, BGPDT, for the specified boundary condition.
BCBULK	ENGINEERING	Builds boundary condition dependent matrices, transfer functions and initial conditions.
BLSTDRV	ENGINEERING	Performs the response of an aircraft to a nuclear blast. Computes modal displacements, velocities, and accelerations.
BLASTFIT	ENGINEERING	Computes the interpolated time domain steady state and time dependent unsteady aerodynamic influence coefficients for blast analyses.
BLASTRIM	ENGINEERING	Performs a trim analysis of an aircraft in order to establish initial conditions for a nuclear blast response calculation.
BOUND	ENGINEERING	Returns flags to the MAPOL sequence that define the matrix reduction path for the current boundary condition.

Table 13. Summary of ASTROS Modules — Continued

MODULE NAME	TYPE	DESCRIPTION
CEIG	ENGINEERING	Computes the complex eigenvalues and eigenvectors of a matrix.
COLMERGE	MATRIX	Merges two or more submatrices into a single matrix based on column partitioning vectors.
COLPART	MATRIX	Partitions a matrix into two or more submatrices based on column partitioning vectors.
DCEVAL	ENGINEERING	Evaluates displacement constraints in the current boundary condition.
DDLOAD	ENGINEERING	Computes the sensitivities of design dependent loads for active boundary conditions.
DECOMP	MATRIX	Decomposes a matrix into its triangular factors.
DESIGN	ENGINEERING	Performs redesign by math programming methods based on the current set of active constraints and constraint sensitivities.
DESPUNCH	UTILITY	Writes new modified Bulk Data entries for the current design iteration to the PUNCH file.
DMA	ENGINEERING	Assembles the direct and/or modal stiffness, mass and/or damping matrices including extra point degrees of freedom for dynamic analysis disciplines.
DYNLOAD	ENGINEERING	Assembles the direct and/or modal time and/or frequency dependent loads including extra point degrees of freedom for dynamic response disciplines.
DYNRSP	ENGINEERING	Computes the direct or modal displacements, velocities and accelerations for transient, frequency and blast analyses.
EDR	ENGINEERING	Computes the stresses, strains, grid point forces and strain energies for elements selected for output for the particular boundary condition.
EMA1	ENGINEERING	Assembles the element stiffness and mass matrices (stored in the KELM and MELM entities) into the design sensitivity matrices DKVI, DMVI.
EMA2	ENGINEERING	Assembles the element stiffness and mass matrix sensitivities (stored in the DKVI and DMVI entities) into the global stiffness and mass matrices for the current design iteration.
EMG	ENGINEERING	Computes the element stiffness, mass, thermal load and stress component sensitivities for all structural elements.
EXIT	UTILITY	Terminates the execution of the MAPOL sequence. Useful to terminate modified MAPOL sequences.
FBS	MATRIX	Performs the forward-backward substitution to solve systems of linear equations.
FCEVAL	ENGINEERING	Evaluates the current value of all frequency constraints.
FLUTDMA	ENGINEERING	Assembles the dynamic matrices for the FLUTTER disciplines.
FLUTDRV	ENGINEERING	Driver for FLUTTER analyses.
FLUTQHHL	ENGINEERING	Processes the {QKKL} matrix with normal modes for FLUTTER.
FLUTSENS	ENGINEERING	Computes the sensitivities of active flutter constraints in the current active boundary condition.
FLUTTRAN	ENGINEERING	Performs flutter analyses in the current boundary condition and evaluates any flutter constraints if it is an optimization boundary condition with applied flutter constraints.

Table 13. Summary of ASTROS Modules — Continued

MODULE NAME	TYPE	DESCRIPTION
FREDUCE	ENGINEERING	Reduces the symmetric or asymmetric f-set stiffness, mass and/or loads matrix to the a-set if there are omitted degrees of freedom.
FREQSENS	ENGINEERING	Computes the sensitivities of active frequency constraints in the current active boundary condition.
FSD	ENGINEERING	Performs redesign by fully stressed design methods based on the set of applied stress constraints. All other applied constraints are ignored.
GDR1	ENGINEERING	Computes the shifted stiffness matrix and the rigid body transformation matrix [GGO] to be used in Phase 2 of Generalized Dynamic Reduction.
GDR2	ENGINEERING	Computes the orthogonal basis [PHI0K] for the general Krylov subspace to be used in Phase 3 of Generalized Dynamic Reduction.
GDR3	ENGINEERING	Computes the transformation matrix [GSUBO] for Generalized Dynamic Reduction.
GDR4	ENGINEERING	Computes transformations between displacement sets useful for data recovery from Generalized Dynamic Reduction.
GPWG	ENGINEERING	Grid point weight generator module.
GREduce	ENGINEERING	Reduces the symmetric g-set stiffness, mass or loads matrix to the n-set if there are multipoint constraints in the boundary condition.
GTLOAD	ENGINEERING	Assembles the current static applied loads matrix for any statics subcases in the current boundary condition from the constant simple load vectors and the design dependent load sensitivities.
IFP	ENGINEERING	Reads the Bulk Data File and loads the input data to relations. Computes the external coordinate system transformation matrices and creates the basic grid point data. Also performs bandwidth minimization.
INERTIA	ENGINEERING	Computes the rigid body accelerations for statics analyses with inertia relief.
ITERINIT	ENGINEERING	Initializes the CONST relation for the current iteration.
LODGEN	ENGINEERING	Assembles the simple load vectors and simple load sensitivities for all applied loads in the Bulk Data File.
MAKDFU	ENGINEERING	Assembles the sensitivities to the displacements of active stress and displacement constraints in the current active boundary condition.
MAKDFV	ENGINEERING	Assembles the sensitivities of active thickness constraints.
MAKDVU	ENGINEERING	Multiplies the stiffness or mass design sensitivities by the active displacements or accelerations.
MAKEST	ENGINEERING	Generates the element summary relational entities for all structural elements. Determines the design variable linking and generates sensitivities for any thickness constraints.
MERGE	MATRIX	Merges two or more submatrices into a single matrix based on row and column partitioning vectors.
MKAMAT	ENGINEERING	Assembles the constraint sensitivity matrix from the sensitivity matrices formed by MAKDFU and the sensitivities of the displacements for active static load conditions in the current active boundary condition.

Table 13. Summary of ASTROS Modules — Continued

MODULE NAME	TYPE	DESCRIPTION
MKUSET	ENGINEERING	Generates the structural set definition entity USET for each boundary condition and forms the partitioning vectors and transformation matrices used in matrix reduction.
MK2GG	ENGINEERING	Interprets solution control and generates the (M2GG) and (K2GG) matrices if necessary.
NREDUCE	ENGINEERING	Reduces the symmetric n-set stiffness, mass or loads matrix to the f-set if there are single point constraints in the boundary condition.
NULLMAT	ENGINEERING	Breaks data base equivalences from previous boundary conditions.
OFFAEROM	ENGINEERING	Solves for the SAERO applied loads and displacements on aero boxes for output requests.
OFFDISP	ENGINEERING	Prints selected displacements, velocities and/or accelerations from any analyses in the current boundary condition.
OFFALOAD	ENGINEERING	Solves for the SAERO applied loads and constraint forces for output processing.
OFFDLOAD	ENGINEERING	Processes output requests for dynamics loads (transient frequency, and gust).
OFFEDR	ENGINEERING	Prints selected element stress, strain, force and/or strain energies from any analyses in the current boundary condition.
OFFGRAD	ENGINEERING	Processes output requests for objective and constraint gradients.
OFFLOAD	ENGINEERING	Prints selected applied external loads from any analyses in the current boundary condition.
OFFMROOT	ENGINEERING	Processes output requests for normal modes.
OFFSPCF	ENGINEERING	Processes output requests for single-point constraint forces.
PARTN	MATRIX	Partitions a matrix into two or more submatrices based on row and column partitioning vectors.
PFBULK	ENGINEERING	Performs a number of preface operations to form additional collections of data.
QHHGEN	ENGINEERING	Computes the discipline dependent unsteady aerodynamic matrices for GUST and BLAST analyses in the modal structural system.
RECEAD	CADDB	Terminates setting conditions on a MAPOL relational access.
RECOVA	ENGINEERING	Recovers the symmetric or asymmetric f-set displacements or accelerations if there are omitted degrees of freedom.
REIG	ENGINEERING	Computes the eigenvalues and eigenvectors of the system as directed by the boundary METHOD selection.
RELCND	CADDB	Sets conditions on attribute values for MAPOL retrieval of relational entities.
RELADD	CADDB	Adds a tuple to an entity opened with RELUSE .
RELEND	CADDB	Closes an entity opened from the MAPOL sequence using RELUSE .
RELGET	CADDB	Retrieves a relational tuple into execution memory for a relation opened for use in the MAPOL sequence.
RELUPD	CADDB	Performs a relational update from execution memory of a tuple retrieved using RELGET .
RELUSE	CADDB	Opens a relational entity for access from the executive sequence.
ROWMERGE	MATRIX	Merges two or more submatrices into a single matrix based on row partitioning vectors.

Table 13. Summary of ASTROS Modules — Continued

MODULE NAME	TYPE	DESCRIPTION
ROWPART	MATRIX	Partitions a matrix into two or more submatrices based on row partitioning vectors.
SAERO	ENGINEERING	Solves the trim equation for steady aeroelastic trim analyses. Computes the rigid and flexible stability coefficients for steady aeroelastic analyses and the aerodynamic effectiveness constraints for constrained optimization steady aerodynamic analyses.
SAERODRV	ENGINEERING	Driver for SAERO disciplines.
SAEROMRG	ENGINEERING	Merges the SAERO results into [MATOUT] in case order.
SCEVAL	ENGINEERING	Computes the stress and/or strain constraint values for the statics or steady aeroelastic trim analyses in the current boundary condition.
SOLUTION	ENGINEERING	Interprets the solution control packet, forms the CASE entity and outputs certain key parameters to the executive sequence.
SPLINES	ENGINEERING	Generates interpolation matrix relating displacements and forces between the steady aero and structural models.
SPLINEU	ENGINEERING	Generates interpolation matrix relating displacements and forces between the unsteady aero and structural models.
STEADY	ENGINEERING	Computes rigid unit forces and aeroelastic corrections for steady aero.
STEADYNP	ENGINEERING	Computes rigid trimmed forces for non-planar models.
TCEVAL	ENGINEERING	Computes the current values of thickness constraints for this optimization iteration.
TRNSPOSE	MATRIX	Transposes a matrix.
UNSTEADY	ENGINEERING	Computes unsteady generalized forces.
USETPRT	UTILITY	Prints the structural set definition table from the USET entity for the specified boundary condition.
UTGPRT	UTILITY	Prints several specific matrix entities in an interpretable form.
UTMPRG	UTILITY	Purges matrix entities.
UTMPRT	UTILITY	To print any matrix entity.
UTRPRG	UTILITY	Purges relational entities.
UTRPRT	UTILITY	To print any relational entity. Only the first twelve attributes are printed and character attributes must be eight characters in length or they will be ignored.
UTUPRG	UTILITY	Purges unstructured entities.
UTUPRT	UTILITY	To print any unstructured entity.
VANGO	ENGINEERING	Performs redesign using optimality criteria methods.
YSMERGE	ENGINEERING	A special purpose merge utility for merging YS-like vectors (vectors of enforced displacements) into matrices for data recovery.

2.4.2.2. The Preface Segment

In the context of optimization, it is obvious that invariant data should be computed only once and re-used subsequently for each iteration. This is the underlying principle involved in the determination of which modules constitute preface modules. In each instance, the data generated are invariant with respect to the design variables.

The preface segment begins with a call to the solution control interpreter to determine the number and types of analyses to be performed. The input file processor (IFP) is then called. The element connection data and element matrices are then formed. PFBULK is then called to perform error checking operations on a variety of user input data. If any nonplanar steady aerodynamic analyses are requested in the solution control, they are analyzed next. Then the EMA1 is called to compute the design invariant stiffness and mass sensitivities to the global design variables. Then the simple loads and load sensitivities are computed in LODGEN. If any planar static aerodynamic analyses are requested in the solution control, the STEADY and SPLINES modules are called to create the aerodynamic matrices required for the aeroelastic analysis. Finally, unsteady aerodynamics matrices are computed for BLAST, GUST and FLUTTER analyses in UNSTEADY, AMP and SPLINEU.

2.4.2.3. The Analysis/Optimization Segments

The remainder of the MAPOL algorithm consists of the optimization and analysis segments. Any particular boundary condition is either an optimization boundary condition (implying that the quantities computed in the disciplines selected in the solution control are constrained and that the structure is to be optimized subject to those constraints) or an analysis boundary condition. The design of the ASTROS system requires that all optimization boundary conditions precede any analysis boundary conditions. The analysis segment (labeled the "final analysis") is intended to follow an optimization with analyses in disciplines whose output values are not constrained but are of interest to the designer or to provide the user with an opportunity to view additional output not desired within the optimization loop. Also the analysis segment can be used on a stand alone basis to perform any desired analyses.

Both the optimization and analysis segments consist of an initial loop on the number of boundary conditions. The analyses in these loops support all the disciplines currently available in the ASTROS system and differ only in the respect that the analysis segment does not have calls to constraint evaluation modules and the optimization segment has convergence tests and design iteration initialization outside the analysis boundary condition loop. The first step in these loops is to assemble the boundary condition dependent number of degrees of freedom (extra points are BC selectable in ASTROS). Then additional PFBULK-like operations are performed in BCBULK to ensure that BC-dependent user input is correct. Then the global stiffness and/or mass matrices are assembled and, if needed, the global loads matrix. Following these tasks, there are several BLOCK IF statements on the various dependent structural sets. In executing each block, the required matrix partitions and reductions are performed. Once the reduced matrices have been obtained for the analyses being performed within the loop, the lowest level response quantities (e.g. displacements, eigenvalues, etc.) are computed. Following the solution, the execution proceeds through another group of dependent set BLOCK IF's to recover the solution vectors to the global set. At this point, the analysis segment is completed with calls to the output file processor modules to compute and output high level response quantities (e.g. stresses).

In the optimization phase of the optimization segment, the **ACTCON** module determines the status of the global convergence flag **CONVERGE** and, if the optimization is not complete, the redesign task is performed. Three redesign methods are supported by the standard sequence and selected through the Solution Control. If the option for Fully Stressed Design (**FSD**) is selected, the redesign is performed in the **FSD** modules. The two alternative methods (mathematical programming and generalized optimality criteria) employ methods that require sensitivity information. In this case, the sensitivities of the active constraints (chosen by **ACTCON** based on the **NRFAC** and **EPS** parameters) are computed.

The sensitivities of the active constraints which are explicit functions of the design variables are computed first in the **MAKDFV** module. Then the second boundary condition loop within the optimization segment begins. The **ABOUND** routine determines the types of active constraints in each boundary condition and outputs logic flags to control the subsequent sensitivity computations. Then boundary condition dependent constraints which are explicit functions of the design variables (frequency and flutter) are computed. Next, the sensitivities of the constraints to the displacements for those **STATICS** constraints which are explicit functions of the displacements (e.g., stress and displacement constraints) are computed using the **MAKDFU** module. For these types of constraints, the product of the stiffness sensitivities and the displacements and the mass sensitivities and the accelerations are also computed and modified appropriately to account for design dependent loads and inertia relief. The resulting matrix is then reduced and used to solve for the sensitivities of the displacements to the design variables. This matrix is recovered to the free displacement set in a manner similar to the recovery of the outputs in the analysis phase of the optimization segment. The final module within the boundary condition loop for sensitivity evaluation is **MMAMAT**. Within this module the constraint sensitivities to the design variables are formed from the product of the two sensitivity matrices previously obtained.

For static aeroelastic analyses, a procedure similar to that for **STATICS** is used twice: once for "pseudo-displacements" that allow computation of aeroelastic effectiveness derivatives and once for real displacements that support the static strength constraints. The static aeroelastic sensitivity code is further complicated by the generality of the aeroelastic correction matrix selections, which are subcase dependent.

After all the active optimization boundary conditions have been processed, the **DESIGN** or the **VANGO** module is called. Within these modules, the approximate design problem is arranged for use by the optimizer and is solved. Following convergence of the approximate problem, execution returns to the top of the optimization loop and a complete reanalysis of all the boundary conditions is performed. Once completed, the **ACTCON** module determines if the global problem is converged and, if so, sets the global convergence flag to **TRUE** causing the execution to pass to the top of the analysis segment. If any analysis boundary conditions exist, they will be processed in a manner similar to the analysis phase of the optimization segment. After performing the requested final analyses (if any) the executive system terminates the **ASTROS** execution.

2.4.3. Modifying the Standard MAPOL Sequence

The standard MAPOL sequence is provided to allow a user to run the **ASTROS** system without detailed knowledge of the MAPOL language or the standard sequence. There is not, however, any requirement that the standard sequence be used. Chapter 6 outlines the procedure for writing a valid MAPOL sequence, and any series of syntactically correct MAPOL statements may be used to direct the **ASTROS** procedure. All the engineering, utility and matrix manipulation modules shown in Subsection

2.4.2.1 are available to any MAPOL sequence used to direct the system. In addition, there are a number of intrinsic functions such as **SIN** and **ABS** that are also available. Their use is detailed in the MAPOL Programming chapter. The sophisticated MAPOL user is thus provided with a very flexible control language to manipulate the ASTROS system. This subsection describes simple modifications to the standard algorithm to print out additional data items, to fine tune the optimization algorithm and to restore an ASTROS analysis that was partially executed on a previous run. No set of examples, however, can possibly indicate the full range of available capabilities; the user is therefore cautioned not to be overly constrained by this discussion.

In order to avoid vast quantities of output and to limit the execution time, the standard output is kept to a minimum. Several utilities, listed in Section 2.4.2.1, can, however, be inserted in the standard sequence to output data stored on the data base. In addition, a utility has been written to print out the structural set definition table to aid in the debugging of the structural model. The **UTMPRT**, **UTGPRT**, **UTRPRT** and **UTUPRT** print utilities dump the contents of specified data base entities to the user's output file. These can be used anywhere in the MAPOL sequence after the specified entity has been filled with data. The **USETPRT** utility provides the user with the ability to print the structural set definition table (**USET**) in a format which aids in debugging the structural model. These utilities provide the user with some simple tools to allow closer interaction with the data stored on the data base and to provide capability to more closely track the execution.

The print utilities provide data visibility without modifying the basic execution of the standard sequence. At a slightly more complex level, the user might desire to fine tune the optimization procedure or to track the iterations of the optimizer more closely. Table 4 includes a number of parameters which are used by ASTROS to direct the optimization. All of these parameters can be modified through the **OPTIMIZE** command in solution control. That modification, however, only occurs once. Any of these parameters can be changed by the user at any point in the MAPOL sequence. For example, the **MOVLIM** parameter could be changed to a different value after the fifth iteration by placing the following statement immediately after the **WHILE** test on **GLBCNVRG**:

```
IF NITER > 5 MOVLIM = 1.5;
```

Obviously, the conditional testing could become as complex as the MAPOL programmer desires.

The brief discussion above does not begin to describe all the options open to the sophisticated ASTROS user. It does, however, outline some of the most commonly performed modifications to the standard MAPOL algorithm. The concepts described can be extended to a large number of similar changes; e.g., modifying the input dynamic pressure value within the MAPOL sequence could be done to avoid re-running the base run of an ASTROS execution. At a more advanced level, the MAPOL relational data base entity utilities can be used to directly modify the design variable values or objective sensitivities.

2.4.4. Restart Capability

The ASTROS system has an informal restart capability which is described in the following sections.

2.4.4.1. Introduction

Currently, ASTROS does not support a formal restart capability but this does not imply that restarts cannot be performed in ASTROS. The restart capability in ASTROS is limited in that the user must use a modified MAPOL sequence in order to terminate the system early and the restarted job **MUST** use a tailored MAPOL sequence to restart the job at the desired point. Otherwise, there are no limits to what can be done by the experienced user. The ASTROS restart capability is best described as a full featured **Manual Restart** — ASTROS does not have an **Automated Restart**.

There are several reasons why users may wish to suspend an ASTROS execution and then perform a restart and the code supports this basic capability. For example, a user may wish to examine the progress of a design after each optimization iteration. With the run stopped, the user would then have the freedom to use ICE (Reference 7) and alter ASTROS data to redirect the optimization path if desired. As another example, the user may suspend execution and, again using ICE, replace ASTROS-computed data with their external equivalent (such as the QHHL or QKKL matrices of unsteady aerodynamic influence coefficients). Clearly, the ability to suspend/restart executions in combination with the ICE environment opens limitless possibilities.

Without an automated restart, however, the user is responsible for ensuring that several requisite tasks are completed. These are

- ☞ Ensuring that the run-time database has the proper **STATUS** on suspension and on restart.
- ☞ Selecting where in the MAPOL sequence to suspend execution.
- ☞ Writing a MAPOL sequence to restart execution. This may or may not be a modification to the standard sequence.
- ☞ Ensuring that those scalar variable(s) that are common to both the original and the restart MAPOL sequences are initialized to the correct value(s)

Each of these tasks are discussed in the following sections.

2.4.4.2. Ensuring proper STATUS of the run-time database

When suspending execution, the run-time database must be saved. ASTROS stores all the information that it has generated during the execution on the run-time database and, on any restart, the downstream modules will expect that those data will exist when they are executed in the restart environment. The run-time database is also the location of the data that the user may wish to modify or add to using ICE prior to initiating the restart. Saving the run-time database is done by selecting a **STATUS** of **NEW** (with the optional user parameter, **KEEP**, if required on the local host) on the **ASSIGN DATABASE** entry. For example,

```
ASSIGN DATABASE CALVIN HOBBS NEW
```

```
ASSIGN DATABASE CALVIN HOBBS NEW KEEP
```

When restarting ASTROS using an existing database whose contents are to be preserved, ASTROS must be notified to attach the existing run-time database files without re-initializing them. This is done by selecting a **STATUS** of **OLD** on the **ASSIGN DATABASE** entry. For example

```
ASSIGN DATABASE CALVIN HOBBS OLD
```

If the **STATUS** of **OLD** is not given, existing database files are typically overwritten by the system. The **STATUS** flag indicates the status of the *data* not of the *files* so the files may exist with a **STATUS** of **NEW** and will result in the database contents being replaced by the new execution.

2.4.4.3. Suspending/Restarting Execution

ASTROS execution is controlled by the MAPOL sequence that is supplied in the MAPOL packet. This may be the standard sequence (if the packet is omitted), an edited version of the standard sequence, or a user supplied sequence. To suspend execution, a MAPOL sequence must be defined which results in clean termination (one without fatal errors) of the ASTROS execution. This may be the standard execution or, more typically, an edited standard sequence or even a standalone MAPOL program.

Most commonly, the suspension is performed by editing the standard MAPOL sequence and inserting an **EXIT** call after the last line that is to be executed in the current execution. Alternatively, if no missing **IF-THEN-ENDIFs** or **ENDDOs** result, portions of the sequence can simply be deleted. Some care should be taken in suspending execution in the middle of **DO** and **DO WHILE** loops or block **IFs**. Although possible to do, suspensions during execution of these repetitive segments can leave the system in a state that is more difficult to reinitialize on the restart execution. Some experience with MAPOL and with ASTROS is needed before attempting these more complex suspensions. Suspending execution at the beginning or end of the preface, analysis phase (of either the optimization or final analysis segments) or the sensitivity phase is most likely to yield success.

To restart the execution, the user *must* generate a special MAPOL program either by editing the standard sequence or by writing a new sequence. The restart execution of ASTROS does not have any information on where the initial execution terminated. Only the data on the database is saved (i.e., available for the current execution). Obviously, the new execution may start up at any point the user wishes and need not be associated with the area where the initial run terminated (although only experienced ASTROS users should attempt to drastically alter the flow of the MAPOL sequence).

To generate the special MAPOL sequence, the user may use a **GOTO** statement to jump ahead in the standard sequence to the restart point or the user may use the **EDIT** commands to delete those initial sections that no longer need execution. The latter is typically the case when the preface segment is "saved" for restart. If the user deletes lines, care should be taken not to delete half of a looping construct or block **IF** since that will result in a MAPOL compilation error. The restart MAPOL sequence must also contain any new statements that are required to reset values of MAPOL scalar parameters.

2.4.4.4. Resetting MAPOL Parameters

In the ASTROS system, all the values of MAPOL variables are stored on the database. For the complex data types like RELATIONS and MATRICES, this is obvious, since their data resides on the database for ICE execution or other processing. Less obviously, the simple data types like REAL and INTEGER (including arrays) are also saved on the database. These data are not easily viewed in the ICE context, but are saved in a way that the MAPOL compiler can recognize. When a restart job is performed, the existence of these old data causes the MAPOL compiler to determine the correspondence between the original data and the new MAPOL sequence. Whenever a variable of the same name and type is found, its initial value is recovered from the old data thus "restoring" the value of the original variable to the last value it contained.

In the restart execution, however, the user must make sure that the *last* value of the variable is the desired *initial* value for restart. In some cases, the variables contain "invariants" like the variable NDV which contains the number of global design variables. In other cases, like BC, the variable is a loop counter that should be reset. The MAPOL sequence may perform the reinitialization automatically (for example if a DO loop is re-executed for values 1 through 10, the do loop counter will be reset to 1 no matter what value it contains). If, however, the restart MAPOL omits the loop, the last loop counter that was achieved will be stored in the loop counter on restart.

Determining which MAPOL parameters should be left alone and which should be reset (rather than default to their last value) is the challenge of the manual restart. Tables 7 through 12 (section 2.4.1) of this Manual have a list of all the MAPOL parameters. These tables list each parameter and give a description of how and where the parameter is used. Together with the ASTROS Programmer's Manual, which document the actions that occur in each ASTROS module, the user can decide which parameters should be reset and which should be allowed to default to the value set in the initial execution.

2.5. MAPOL PROGRAM LISTING

The Version 10 MAPOL listing of the standard solution sequence is given on the following pages.

Standard MAPOL Sequence Listing

```

STAT LEVL
1  1!$***$
2  1!$ CSCIID <@(#) MC0083-MAPOLSEQ 10.1 7/1/93 11:27:52> $
3  1!$***$
4  1!$.....$!
5  1!$ EXECUTIVE SEQUENCE FOR ASTROS $!
6  1!$.....$!
7  1!$.....$!
8  1!$ CONSTANTS FOR SDCOMP SET SINGULARITY MESSAGES $!
9  1!$.....$!
10 1!$ INTEGER SINGOSET, SINGASET, SINGLSET; $!
11 1!$.....$!
12 1!$ VARIABLE DECLARATION SEGMENT $!
13 1!$.....$!
14 1!$ $!
15 1!$ INTEGER GSIZE, NDV, NITER, BC, $!
16 1!$ ESIZE(1000), PSIZE(1000), GSIZEB; $!
17 1!$ REAL CTL, CTLMIN; $!
18 1!$ LOGICAL GLBCNVRG, APPCNVRG, PFLAG; $!
19 1!$ UNSTRUCT DCENT, GRIDTEMP, SMPLOD; $!
20 1!$ RELATION DESHIST, CONST, MPPARM, CONVERT, OCPARM, $!
21 1!$ MFORM, GRID, SPOINT, EPOINT, SEQGP, $!
22 1!$ BGPDT(1000), CSTM, FORCE, FORCE1, MOMENT, $!
23 1!$ MOMENT1, PLOAD, GRAV, LOAD, EIGR, $!
24 1!$ TEMP, TEMPD, $!
25 1!$ CORD1C, CORD1R, CORD1S, CORD2C, CORD2R, $!
26 1!$ CORD2S, GPWGGRID, OGPWG, GRADIENT; $!
27 1!$ $!
28 1!$.....$!
29 1!$ DECLARATIONS FOR MODULE MKUSET $!
30 1!$.....$!
31 1!$ $!
32 1!$ UNSTRUCT USET(1000); $!
33 1!$ RELATION SPC, SPC1, SPCADD, MPC, MPCADD, $!
34 1!$ ASET, ASET1, OMIT, OMIT1, SUPORT, $!
35 1!$ JSET, JSET1, RBAR, RBE1, RBE2, RBE3, RR0D; $!
36 1!$ MATRIX [PGMN(1000)], [PNSF(1000)], [PFOA(1000)], [PARL(1000)], [TMN(1000)], $!
37 1!$ [YS(1000)]; $!
38 1!$ $!
39 1!$.....$!
40 1!$ DECLARATIONS FOR MODULES MAKEST AND EMG $!
41 1!$.....$!
42 1!$ $!
43 1!$ UNSTRUCT TREF, DVSIZE, PCOMPS; $!
44 1!$ UNSTRUCT KELM, MELM, TELM; $!
45 1!$ RELATION CQDMEM1, QDMM1EST, CROD, CONROD, RODEST, $!
46 1!$ CSHEAR, SHEAREST, CTRMEM, TRMEMEST, CMAS1, $!
47 1!$ CMAS2, MASSEST, CONM1, CONM1EST, CONM2, $!
48 1!$ CONM2EST, CBAR, BEAMEST, CQUAD4, QUAD4EST, $!
49 1!$ CIHEX1, IHEX1EST, CIHEX2, IHEX2EST, CIHEX3, $!
50 1!$ IHEX3EST, CELAS1, CELAS2, ELASEST, $!
51 1!$ PCOMP, PQDMEM1, PROD, PSHEAR, $!
52 1!$ PTRMEM, PMASS, PELAS, PBAR, PSHELL, $!

```

Standard MAPOL Sequence Listing — Continued

```

STAT LEVL
53 1!          PCOMP1,    PCOMP2,    PIHEX,    MAT1,    MAT2,    !
54 1!          MAT8,      MAT9,      CTRIA3,    TRIA3EST,    !
55 1!$
56 1!$*****$!
57 1!$          DECLARATIONS FOR DESIGN VARIABLES/CONSTRAINTS AND LINKING    $!
58 1!$*****$!
59 1!$
60 1!RELATION  DESELM,    DESVARP,    DESVARS,    PLIST,    ELIST,    !
61 1!          SHAPE,      SHPGEN;    !
62 1!RELATION  DCONVM,    DCONTW,    DCONEP,    DCONFT,    DCONVMM,    !
63 1!          DCONVMM,    DCONEPM,    DCONFTM,    DCONVMP,    DCONTW,    !
64 1!          DCONEPP,    DCONFTP;    !
65 1!RELATION  DCONDSP,    DCONFRQ,    DCONTHK,    DCONTH2;    !
66 1!RELATION  DCONPMN,    DCONLMN,    DCONLAM;    !
67 1!RELATION  GLBDES,    DESLINK,    TFIXED,    LOCLVAR,    DVCT;    !
68 1!MATRIX    [PTRANS];    !
69 1!IMATRIX   [PMINT],    [PMAXT],    [SMAT];    !
70 1!$
71 1!$*****$!
72 1!$          DECLARATIONS FOR OUTPUT FILE PROCESSING (EDR/OF)    $!
73 1!$*****$!
74 1!$
75 1!RELATION  GRIDLIST,    MODELIST,    ELEMLIST,    FREQLIST,    TIMELIST,    !
76 1!          ITERLIST,    GDVLIST,    LDVLIST,    DCONLIST,    PLYLIST;    !
77 1!$
78 1!RELATION  GPPELEM,    EOSUMMRY,    EOBAR,    EOELAS,    EOHEX1,    !
79 1!          EOHEX2,    EOHEX3,    EOQDMM1,    EOQUAD4,    EOROD,    !
80 1!          EOSHEAR,    EOTRMEN,    GPFDATA,    EOTRIA3;    !
81 1!UNSTRUCT  EODISC;    !
82 1!$
83 1!RELATION  OGRIDLOD,    OGRIDDSP,    OLOCALDV,    OAGRDDSP,    OAGRDL0D;    !
84 1!MATRIX    [FLUTMODE],    [PTGLOAD],    [PFGLOAD],    [PTHLOAD],    [PFHLOAD];    !
85 1!$
86 1!$*****$!
87 1!$          DECLARATIONS FOR MODULES EMA1, EMA2 AND GLOBAL    $!
88 1!$          MATRIX PARTITION/REDUCTION    $!
89 1!$*****$!
90 1!$
91 1!UNSTRUCT  DKVI,      DMVI;    !
92 1!RELATION  GMKCT,      GMMCT;    !
93 1!MATRIX    [KGG],      [KNN],      [KFF],      [KAA],      [KLL],      !
94 1!          [MGG],      [MNN],      [MFF],      [MAA],      [MLL],      !
95 1!          [MRRBAR],    [MLR],      [KFS],      [KSS],      [KOOINV(1000)],    !
96 1!          [GSUBO(1000)],    [KLLINV(1000)],    [MRR(1000)],    !
97 1!          [IFM(1000)],    [M1GG],      [IFR(1000)],    [KRR],      [D(1000)],    !
98 1!          [KLR],      [K1GG],      [LHS(1000)],    [M2GG],      [MOO],      !
99 1!          [MOA],      [K2GG],      [MAABAR];    !
100 1!MATRIX    [TMP1],      [TMP2];    !
101 1!MATRIX    [PG],      [PN],      [PF],      [PA],      !
102 1!          [PO],      [PLBAR],    [PR],      [RHS(1000)],    [UG(1000)],    !

```


Standard MAPOL Sequence Listing — Continued

```

STAT LEVEL
103 1!      [UN],      [UF],      [UA],      [UL],      [UM],      !
104 1!      [AG(1000)], [AN],      [AF],      [AA],      [AR],      !
105 1!      [AL],      [UO],      [UOO],     [PS];      !
106 1! LOGICAL  M2GGFLAG, K2GGFLAG;      !
107 1!$      !
108 1!$*****$!
109 1!$      DECLARATIONS FOR SOLUTION CONTROL      $!
110 1!$*****$!
111 1!$      !
112 1! INTEGER  NUMOPTBC, NBNDCOND, MAXITER,      !
113 1!      MPS,      MPE,      !
114 1!      OCS,      OCE,      !
115 1!      FSDS,      FSDE;      !
116 1! INTEGER  BLOAD,      BMASS,      BMODES,      BSAERO,      BFLUTR,      !
117 1!      BDYN,      BDRSP,      BDTR,      BMTR,      BDFR,      !
118 1!      BMFR,      BGUST,      BBLAST,      NMPC,      NSPC,      !
119 1!      NOMIT,      NRSET,      DMODES;      !
120 1! REAL      MOVLIM,      WINDOW,      OCMOVLIM,      ALPHA,      CNVRGLIM,      !
121 1!      NRFAC,      EPS;      !
122 1! RELATION JOB,      OPTIMIZE,      CASE;      !
123 1!$      !
124 1!$*****$!
125 1!$      DECLARATIONS FOR SENSITIVITY EVALUATION      $!
126 1!$*****$!
127 1!$      !
128 1! INTEGER  DDFLG,      NACSD,      NAUS,      NAUA;      !
129 1! LOGICAL  ACTBOUND, ACTFLUT, ACTDYN, ACTAERO, ACTAEFF,      !
130 1!      ACTUAG,      ACTUAGG;      !
131 1! UNSTRUCT PCAS,      PCAE;      !
132 1! MATRIX  [DFDU],      [PGAS],      [UGA],      [DUG],      [DMUG],      !
133 1!      [DPFV],      [DPOV],      [DPNV],      [DPAV],      [DUAV],      !
134 1!      [DUAD],      [DUFV],      [AGA],      [AMAT],      [DKUG],      !
135 1!      [DPGV],      [DPLV],      [DURD],      [DULD],      [DULV],      !
136 1!      [DDELVD], [DPRV],      [DRHS],      [DFDUF],      [PGAA],      !
137 1!      [DFDUN],      [DMAG],      [DMUN],      [DMUF],      [DMUA],      !
138 1!      [DMUO],      [DMUL],      [DMUR],      [DMU],      [DP1],      !
139 1!      [DK1V],      [AUAGC],      [DURV],      [EFFSENS],      [DU1L],      !
140 1!      [DU1R],      [DU2],      [LHSL],      [LHSU],      [PGAU];      !
141 1! IMATRIX  [GLBSIG],      [DPTHVI],      [DPGRVI],      [DPVJ];      !
142 1!$      !
143 1!$*****$!
144 1!$      AERODYNAMIC ENTITIES      $!
145 1!$*****$!
146 1!$      !
147 1! INTEGER  SYM,      MINDEX,      SUB,      S;      !
148 1! REAL      QDP,      MACH;      !
149 1! LOGICAL  LOOP,      AEFLG(1000), NONPONLY;      !
150 1! UNSTRUCT ACPT,      UNMK;      !
151 1! RELATION AESURF,      AIRFOIL,      AEROS,      AEFACT,      AXSTA,      !
152 1!      BODY,      SPLINE1,      SET1,      SET2,      ATTACH,      !
153 1!      TRIM,      AERO,      BLAST,      CAERO6,      PAERO6,      !
154 1!      GEOMSA,      AECOMPS,      STABCF,      CAERO1,      PAERO1,      !

```

Standard MAPOL Sequence Listing — Continued

```

STAT  LEVEL
155  1!      CAERO2,      PAERO2,      MKAERO1,      MKAERO2,      FLUTTER,      !
156  1!      FLFACT,      CLAMBDA,      DCONALE,      DCONCLA,      DCONFLT,      !
157  1!      CONEFFS,      CONLINK,      GEOMUA,      AECOMPU,      SPLINE2,      !
158  1!      CONEFF,      DCONTRM,      DCONSCF,      AEROGEO,      CAROGEOM,      !
159  1!MATRIX [AIRFR(1000)], [AICMAT(1000)], [AAICMAT(1000)], !
160  1!      [AICS],      [KAFF],      [PAF],      [KAAA],      [PAA],      !
161  1!      [GASUBO(30,33)], [SKJ],      [DIJK],      [D2JK],      !
162  1!      [KARL],      [R11],      [K21(30,33)], [PARBAR], [PAL],      !
163  1!      [PAR(30,33)], [K1112(30,33)], [AIRFORCE], [K22],      !
164  1!      [GTKG],      [GTKN],      [GTKF],      [GSTKG],      [GSTKN],      !
165  1!      [GSTKF],      [GSKF],      [UGTKG],      [UGTKN],      [UGTKF],      !
166  1!      [UGTKA],      [UGTKO],      [UGTKAB], [AITD],      [KARR],      !
167  1!      [R12(30,33)], [R22],      [R32(30,33)], [K11],      [K12(30,33)], !
168  1!      [P1],      [R21(30,33)], [R31(30,33)], [RL11(30,33)], !
169  1!      [RU11(30,33)], [P2],      [MAAA],      [IFMA(30,33)], !
170  1!      [R13(30,33)], [R33],      [DELC],      [PRIGID],      !
171  1!      [AARC],      [AAR],      [AAA(1000)], [UAA(1000)], [AAAGC],      !
172  1!      [PAO(1000)], [AAFTMP], [UAFTMP], [UAN],      [AAN],      !
173  1!      [UAG(1000)], [AAG(1000)], [AAL],      [AAF],      [UAF],      !
174  1!      [KOOL(30,33)], [KOOU(30,33)], [LHSA(30,33)], !
175  1!      [POARO(30,33)], [KAO(30,33)], [UAR],      [RHS(30,33)], !
176  1!      [DELTA(1000)], [PAOC(1000)], [UAAC(1000)], [AAAC(1000)], !
177  1!      [UAFC(1000)], [UANC(1000)], [UAGC(30,33)], [AAFC(1000)], !
178  1!      [AANC(1000)], [AAGC(30,33)], [KL11(30,33)], [KU11(30,33)], !
179  1!      [R11DPL], [R11PAL(30,33)], [R1112(30,33)], !
180  1!      [R1113(30,33)], [UAL],      !
181  1!MATRIX [AJJTL], [QJL], [QKKL], [QHHL],      !
182  1!$      $!
183  1!$      $!
184  1!$      DYNAMIC RESPONSE DECLARATIONS      $!
185  1!$      $!
186  1!$      $!
187  1!INTEGER HSIZE(1000);      !
188  1!UNSTRUCT TFDATA,      ICDATA,      UDLOLY;      !
189  1!RELATION LAMBDA,      OEIGS,      DLONLY,      DLOAD,      TABLED1,      !
190  1!      IC,      TLOAD1,      TLOAD2,      RLOAD1,      RLOAD2,      !
191  1!      TSTEP,      VSDAMP,      TABDMP1,      DLAGS,      TF,      !
192  1!      DMIG,      GUST,      FREQ,      FREQ1,      FREQ2,      !
193  1!      FFT,      FLUTREL;      !
194  1!MATRIX [PHIKH], [QHJL], [QKJL], [PHIA], [MII],      !
195  1!      [PHIO], [PHIF], [PHIN], [PHIG(1000)], [KHHT],      !
196  1!      [KHHF], [BHH], [MHH], [PDT], [PDF],      !
197  1!      [KDDT], [KDDF], [BDD], [MDD], [ICMATRIX],      !
198  1!      [UTRANA], [UFREQA], [UTRANI], [UFREQI], [UFREQE],      !
199  1!      [UTRANE], [UTRANF], [UFREQF], [UTRANN], [UFREQN],      !
200  1!      [UTRANG], [UFREQG], [MHHFL(30,33)], [BHHFL(30,33)], !
201  1!      [QHHLFL(30,33)], [KHHFL(30,33)];      !

```

Standard MAPOL Sequence Listing — Continued

```

STAT LEVL
202 1!$
203 1!$*****$!
204 1!$ DECLARATIONS FOR GENERALIZED DYNAMIC REDUCTION (GDR) $!
205 1!$*****$!
206 1!$ $!
207 1!INTEGER LKSET, LJSET, NEIV, GNORM, NGDR, !
208 1! ASIZE, LSIZE; !
209 1!REAL FMAX; !
210 1!RELATION DYNRED; !
211 1!MATRIX {PGDRG(1000)}, {PHIOK}, {KOO}, {GGO}, {KSOO}, !
212 1! {KOA}, {LSOO}, {PAJK}, {PFJK}, {UFGDR}, !
213 1! {AFGDR}, {UJK}, {GTMP}; !
214 1!$ $!
215 1!$*****$!
216 1!$ BLAST RESPONSE DECLARATIONS $!
217 1!$*****$!
218 1!$ $!
219 1!REAL BQDP; !
220 1!MATRIX {MPART}, {ID2}, {PHIE}, {PHIR}, {PHIB}, !
221 1! {GENM}, {GENK}, {GENF}, {GENQ}, {GENQL}, !
222 1! {DTSPL}, {FTF}, {QRE}, {QEE}, {KEQE}, !
223 1! {LKQ}, {UKQ}, {GFR}, {GFE}, {BTEM}, !
224 1! {BLSTJA}, {BLGTJA}, {BFRC}, {MATTR}, {MATSS}, !
225 1! {KEE}, {DELB}, {DELM}, {URDB}, {GENFA}, !
226 1! {DWNWSH}, {ELAS}, {SLPMOD}, {QRR}, {UBLASTI}, !
227 1! {UBLASTG}, {UBLASTF}; !
228 1!$ $!
229 1!$*****$!
230 1!$ $!
231 1!$ BEGIN MAPOL SOLUTION SEQUENCE $!
232 1!$ $!
233 1!$*****$!
234 1!$ PREFACE MODULES $!
235 1!$*****$!
236 1!SINGOSET := 1; !
237 1!SINGASET := 2; !
238 1!SINGLSET := 3; !
239 1!$ $!
240 1!$ INITIALIZE SUBSCRIPT VALUES TO "1" TO AVOID RUN TIME PROBLEMS $!
241 1!$ $!
242 1!SUB := 1; !
243 1!PRINT("LOG=('BEGIN PREFACE MODULES')"); !
244 1!CALL SOLUTION ( NUMOPTBC, NBNDCOND, MPS, MPE, OCS, OCE, FSDE, FSDE, !
245 1! MAXITER, MCVLIM, WINDOW, OCMOVLIM, ALPHA, CNVRGLIM, !
246 1! NRFAC, EPS ); !
247 1!CALL IFP ( GSIZEB ); !
248 1!$ $!
249 1!$ GENERATE THE ELEMENT MATRICES $!
250 1!$ $!
251 1!PRINT("LOG=('ELEMENT MATRIX GENERATION')"); !
252 1!$ $!

```

Standard MAPOL Sequence Listing — Continued

```

STAT LEVL
253 1!CALL MAKEST ( NDV, GLBDES, [PTRANS], [PMINT], [PMAXT], LOCLVAR,      !
254 1!          TFIXED, DESLINK );      !
255 1!$      $!
256 1!CALL EMG ( NDV, GSIZEB, GLBDES, DESLINK, [SMAT], DVCT, DVSIZE,      !
257 1!          KELM, MELM, TELM, TREF );      !
258 1!$      $!
259 1!CALL PFBULK ( GSIZEB, EOSUMRY, EODISC, GPFELEM );      !
260 1!$      $!
261 1!$          HANDLE THE NON-PLANAR STEADY AERODYNAMICS ANALYSES      $!
262 1!$          TERMINATE THE EXECUTION IF THE ONLY DISCIPLINE IS NPSAERO      $!
263 1!$      $!
264 1!PRINT('LOG=('NON-PLANAR STEADY AERODYNAMICS')');      !
265 1!CALL STEADYNP ( NONPONLY, AECOMPS, GEOMSA, STABCF, [AIRFORCE], AEROGEOM,      !
266 1!          CAROGEOM, OAGRDLOD );      !
267 1!IF NONPONLY CALL EXIT;      !
268 1!$      $!
269 1!$          ASSEMBLE THE ELEMENT MATRICES      $!
270 1!$          TO THE SENSITIVITY MATRICES      $!
271 1!$      $!
272 1!PRINT('LOG=('PHASE 1 ELEM. MATRIX ASSEMBLY')');      !
273 1!CALL EMAL ( NDV, GLBDES, DVCT, KELM, MELM, GMKCT, DKVI, GMMCT, DMVI );      !
274 1!$      $!
275 1!$          GENERATE THE SIMPLE LOAD VECTORS      $!
276 1!$          AND LOAD SENSITIVITIES      $!
277 1!$      $!
278 1!PRINT('LOG=('PHASE 1 STATIC LOADS GENER.')');      !
279 1!CALL LODGEN ( GSIZEB, GLBDES, DVCT, DVSIZE, GMMCT, DMVI, TELM, TREF,      !
280 1!          SMPLOD, [DPTRVI], [DPGRVI] );      !
281 1!$      $!
282 1!$          GENERATE THE STEADY AIC MATRIX AND THE      $!
283 1!$          STEADY SPLINE TRANSFORMATION MATRICES      $!
284 1!$      $!
285 1!PRINT('LOG=('STEADY AERODYNAMICS')');      !
286 1!LOOP := TRUE;      !
287 1!MINDEX := 0;      !
288 1!WHILE LOOP DO      !
289 2!  MINDEX := MINDEX + 1;      !
290 2!  CALL STEADY ( MINDEX, LOOP, AECOMPS, GEOMSA, STABCF, [AICMAT(MINDEX)],      !
291 2!          [AAICMAT(MINDEX)], [AIRFRC(MINDEX)], AEROGEOM, CAROGEOM );      !
292 2!ENDDO;      !
293 1!CALL SPLINES ( GSIZEB, GEOMSA, AECOMPS, AEROS, [GTKG], [GSTKG] );      !
294 1!$      $!
295 1!$          GENERATE THE UNSTEADY AIC MATRIX AND THE      $!
296 1!$          UNSTEADY SPLINE TRANSFORMATION MATRIX      $!
297 1!$      $!
298 1!PRINT('LOG=('UNSTEADY AERODYNAMICS')');      !
299 1!CALL UNSTEADY ( GEOMUA, AECOMPU, [AJJTL], [D1JK], [D2JK], [SKJ] );      !
300 1!CALL AMP ( [AJJTL], [D1JK], [D2JK], [SKJ], [QKKL], [QKJL], [QJJL] );      !
301 1!CALL SPLINEU ( GSIZEB, GEOMUA, AECOMPU, AERO, [UGTRG] );      !
302 1!$      $!
303 1!$.....$!
304 1!$          BEGIN OPTIMIZATION LOOP      $!

```

Standard MAPOL Sequence Listing — Continued

```

STAT LEVL
305 1:$*****$!
306 1:$!
307 1:IF NUMOPTBC > 0 THEN!
308 2: PRINT("LOG=('*****')");!
309 2: PRINT("LOG=('BEGIN OPTIMIZATION')");!
310 2:$!
311 2:$ INITIALIZE MAPOL PARAMETERS$!
312 2:$!$!
313 2: GLBCNVRG := FALSE;!
314 2: APPCNVRG := FALSE;!
315 2:$!$!
316 2:$ BEGIN CONVERGENCE LOOP$!
317 2:$!$!
318 2: WHILE NOT GLBCNVRG AND NITER <= MAXITER DO!
319 3:$!$!
320 3:$ ASSEMBLE THE GLOBAL MATRICES$!
321 3:$!$!
322 3: NITER := NITER + 1;!
323 3: PRINT("LOG=('-----')");!
324 3: PRINT("LOG=(' DESIGN ITERATION ',I3)',NITER);!
325 3: CALL ITERINIT ( NITER, CONST );!
326 3: CALL UTMPRG ( [GLBSIG] );!
327 3: CALL TCEVAL ( NITER, NDV, MOVLIM, WINDOW, GLBDES, LOCLVAR, [PMINT],!
328 3: [PMAXT], TFIXED, CONST );!
329 3: CALL LAMINCON ( NITER, NDV, DCONLAM, DCONLMN, DCONPMN, TFIXED, GLBDES,!
330 3: LOCLVAR, [PTRANS], CONST );!
331 3: CALL EMA2 ( NITER, NDV, GSIZEB, GLBDES, GMKCT, DKVI, [K1GG],!
332 3: GMCT, DMVI, [M1GG] );!
333 3:$!$!
334 3:$ BEGIN BOUNDARY CONDITION LOOP FOR OPTIMIZATION$!
335 3:$!$!
336 3: FOR BC = 1 TO NUMOPTBC DO!
337 4: PRINT("LOG=(' BOUNDARY CONDITION ',I3)',BC);!
338 4:$!$!
339 4:$ ESTABLISH THE BASE USET AND PARTITIONING DATA FOR THE BC$!
340 4:$ THIS DATA MUST BE RECREATED EACH ITERATION SINCE GDR CAN CHANGE IT$!
341 4:$!$!
342 4: CALL MKUSET( BC, GSIZEB, [YS(BC)], [TMN(BC)], [PGMN(BC)], [PNSP(BC)],!
343 4: [PFOA(BC)], [PARL(BC)], USET(BC) );!
344 4:$!$!
345 4:$ MAKE B.C.-DEPENDENT BGPDT FROM BASE, ADDING THE EXTRA POINTS FOR$!
346 4:$ THIS B.C.$!
347 4:$!$!
348 4: CALL BCBGPD( BC , GSIZEB , BGPDT(BC) , ESIZE(BC) );!
349 4: GSIZE := GSIZEB;!
350 4: PSIZE(BC) := ESIZE(BC) + GSIZE;!
351 4:$!$!
352 4:$ PROCESS MATRICES, TRANSFER FUNCTIONS, AND INITIAL CONDITIONS FOR$!
353 4:$ THIS B.C.$!
354 4:$!$!
355 4: CALL BCBULK( BC , PSIZE(BC) , BGPDT(BC) , USET(BC) );!
356 4:$!$!

```

Standard MAPOL Sequence Listing — Continued

```

STAT LEVL
357 4! CALL BOUND ( BC, GSIZE, ESIZE(BC), USET(BC), BLOAD, BMASS, DMODES, !
358 4! BMODES, BSAERO, BFLUTR, BDYN, BDRSP, BDTR, BMTR, BDFR, !
359 4! BMFR, BGUST, BBLAST, NMPC, NSPC, NOMIT, NRSET, NGDR ); !
360 4!$ !
361 4!$ DETERMINE IF ANY M2GG/K2GG INPUT DATA ARE TO BE ADDED $!
362 4!$ !
363 4! CALL NULLMAT ( [KGG], [MGG] ); !
364 4! CALL MK2GG ( BC, GSIZEB, [M2GG], M2GGFLAG, [K2GG], K2GGFLAG ); !
365 4! IF M2GGFLAG THEN !
366 5! [MGG] := [M1GG] + [M2GG]; !
367 5! ELSE !
368 5! [MGG] := [M1GG]; !
369 5! ENDIF; !
370 4! IF K2GGFLAG THEN !
371 5! [KGG] := [K1GG] + [K2GG]; !
372 5! ELSE !
373 5! [KGG] := [K1GG]; !
374 5! ENDIF; !
375 4!$ !
376 4!$ CALL THE GRID POINT WEIGHT GENERATOR FOR THIS BOUNDARY CONDITON $!
377 4!$ !
378 4! CALL GPWG ( NITER, BC, GPWGGRID, [MGG], OGPWG ); !
379 4!$ !
380 4! IF BLOAD <> 0 CALL GTLOAD (NITER, BC, GSIZE, BGPDT(BC), GLBDES, !
381 5! SMPLOD, [DPTHVI], [DPGRVI], [PG], OGRIDLOD); !
382 4!$ !
383 4!$ PARTITION-REDUCTION OF GLOBAL MATRICES $!
384 4!$ !
385 4! IF NUMOPTBC > 1 CALL NULLMAT ( [KNN], [PN], [MNN], !
386 5! [GTKN], [GSTKN], [UGTKN] ); !
387 4! IF NMPC <> 0 THEN !
388 5!$ !
389 5!$ PERFORM MPC REDUCTION $!
390 5!$ !
391 5! PRINT("LOG=(' MPC REDUCTION')"); !
392 5! CALL GREduce ( [KGG], [PG], [PGMN(BC)], [TMN(BC)], [KNN], [PN] ); !
393 5! IF BMAS <> 0 CALL GREduce ( [MGG], [PGMN(BC)], [TMN(BC)], [MNN] ); !
394 5! IF BSAERO <> 0 THEN !
395 6! CALL GREduce (, [GTKG], [PGMN(BC)], [TMN(BC)],, [GTKN]); !
396 6! CALL GREduce (, [GSTKG], [PGMN(BC)], [TMN(BC)],, [GSTKN]); !
397 6! ENDIF; !
398 5! IF BFLUTR <> 0 OR BGUST <> 0 OR BBLAST <> 0 !
399 6! CALL GREduce (, [UGTKG], [PGMN(BC)], [TMN(BC)],, [UGTKN] ); !
400 5! ELSE !
401 5!$ !
402 5!$ NO MPC REDUCTION $!
403 5!$ !
404 5! [KNN] := [KGG]; !
405 5! IF BLOAD <> 0 [PN] := [PG]; !
406 5! IF BMAS <> 0 [MNN] := [MGG]; !
407 5! IF BSAERO <> 0 THEN !
408 6! [GTKN] := [GTKG]; !

```

Standard MAPOL Sequence Listing — Continued

```

STAT  LEVEL
409    6!      [GSTKN] := [GSTKG];                                !
410    6!      ENDIF;                                           !
411    5!      IF BFLUTR <> 0 OR BGUST <> 0 OR BBLAST <> 0 [UGTKN] := [UGTKG];    !
412    5!      ENDIF;                                           !
413    4!      IF NUMOPTBC > 1 CALL NULLMAT ( [KFF], [PF], [MFF], [GTKF], [GSTKF],    !
414    5!      [UGTKF] );                                         !
415    4!      IF NSPC <> 0 THEN                                    !
416    5!$      $!
417    5!$      PERFORM SPC REDUCTION                                $!
418    5!$      $!
419    5!      PRINT("LOG=('          SPC REDUCTION')");          !
420    5!      CALL NREDUCE ( [KNN], [PN], [PNSF(BC)], [YS(BC)], [KFF], [KFS],    !
421    5!      [KSS], [PF], [PS] );                                !
422    5!      IF BMASS <> 0 CALL NREDUCE ( [MNN], , [PNSF(BC)], , [MFF] );          !
423    5!      IF BSAERO <> 0 THEN                                    !
424    6!      CALL NREDUCE ( , [GTKN], [PNSF(BC)], , , , [GTKF] );          !
425    6!      CALL NREDUCE ( , [GSTKN], [PNSF(BC)], , , , [GSTKF] );          !
426    6!      ENDIF;                                           !
427    5!      IF BFLUTR <> 0 OR BGUST <> 0 OR BBLAST <> 0          !
428    6!      CALL NREDUCE ( , [UGTKN], [PNSF(BC)], , , , [UGTKF] );          !
429    5!      ELSE                                              !
430    5!$      $!
431    5!$      NO SPC REDUCTION                                $!
432    5!$      $!
433    5!      [KFF] := [KNN];                                    !
434    5!      IF BLOAD <> 0 [PF] := [PN];                            !
435    5!      IF BMASS <> 0 [MFF] := [MNN];                            !
436    5!      IF BSAERO <> 0 THEN                                    !
437    6!      [GTKF] := [GTKN];                                    !
438    6!      [GSTKF] := [GSTKN];                                    !
439    6!      ENDIF;                                           !
440    5!      IF BFLUTR <> 0 OR BGUST <> 0 OR BBLAST <> 0 [UGTKF] := [UGTKN];    !
441    5!      ENDIF;                                           !
442    4!$      $!
443    4!      IF NUMOPTBC > 1 CALL NULLMAT ( [KAA], [PA], [MAA],    !
444    5!      [KAAA], [PAA], [UGTKA] );                            !
445    4!$      $!
446    4!      IF NGDR <> 0 THEN                                    !
447    5!$      $!
448    5!$      PERFORM THE GENERAL DYNAMIC REDUCTION WHICH IS DISCIPLINE    $!
449    5!$      INDEPENDENT. THE RESULTING [GSUBO] MATRIX WILL BE USED BY    $!
450    5!$      ALL DISCIPLINES                                $!
451    5!$      $!
452    5!      PRINT("LOG=('          DYNAMIC REDUCTION')");          !
453    5!$      $!
454    5!$      OBTAIN THE OMITTED DOF PARTITION OF KFF AND MPF    $!
455    5!$      $!
456    5!      CALL PARTN ( [KFF], [KOO], , [KOA], , [PFOA(BC)] );          !
457    5!      CALL PARTN ( [MFF], [MOO], , , , [PFOA(BC)] );          !
458    5!      ASIZE := GSIZE - NMPC - NSPC - NOMIT;                !
459    5!      LSIZE := ASIZE - NRSET;                                !
460    5!      CALL GDR1 ( [KOO], [MOO], [KSOO], [GGO], LKSET, LJSET, NEIV,    !

```

Standard MAPOL Sequence Listing — Continued

```

STAT LEVL
461 5!                                     FMAX, BC, BGPDT(BC), USET(BC), NOMIT, LSIZE );      !
462 5!$                                     $!
463 5!$                                     LKSET      MEANING      $!
464 5!$                                     <> 0      APPROX. MODE SHAPES SELECTED      $!
465 5!$                                     = 0      NO APPROX. MODE SHAPES IN GDR      $!
466 5!$                                     $!
467 5!      IF LKSET <> 0 THEN      !
468 6!      CALL SDCOMP ( [KSOO], [LSOO], USET(BC), SINGOSET );      !
469 6!      CALL GDR2 ( [LSOO], [MOO], [PHIOK], LKSET, LJSET,      !
470 6!      NEIV, FMAX, BC );      !
471 6!      ENDIF;      !
472 5!      CALL GDR3 ( [KOO], [KOA], [MGG], [PHIOK], [TMN(BC)], [GGO],      !
473 5!      [PGMN(BC)], [PNSF(BC)], [PFOA(BC)], [GSUBO(BC)],      !
474 5!      BGPDT(BC), USET(BC),      !
475 5!      LKSET, LJSET, ASIZE, GNORM, BC );      !
476 5!      CALL GDR4 ( BC, GSIZE, PSIZE(BC), LKSET, LJSET, NUMOPTBC, NBNDCOND,      !
477 5!      [PGMN(BC)], [TMN(BC)], [PNSF(BC)], [PFOA(BC)],      !
478 5!      [PARL(BC)], [PGDRG(BC)], [PAJK], [PFJK], BGPDT(BC),      !
479 5!      USET(BC) );      !
480 5!      ENDIF;      !
481 4!$                                     $!
482 4!      IF BLOAD <> 0 OR BMODES <> 0 OR BFLUTR <> 0 OR BDN <> 0 THEN      !
483 5!$                                     $!
484 5!$      REDUCE THE MATRICES WITHOUT AEROELASTIC CORRECTIONS      $!
485 5!$                                     $!
486 5!      IF NGDR <> 0 THEN      !
487 6!$                                     $!
488 6!$      PERFORM THE GENERAL DYNAMIC REDUCTION      $!
489 6!$                                     $!
490 6!      PRINT('LOG=( SYMMETRIC DYNAMIC REDUCTION )');      !
491 6!$                                     $!
492 6!      [MAA] := TRANS ( [GSUBO(BC)] ) * [ [MFF] * [GSUBO(BC)] ];      !
493 6!      [KAA] := TRANS ( [GSUBO(BC)] ) * [ [KFF] * [GSUBO(BC)] ];      !
494 6!      IF BLOAD <> 0 [PA] := TRANS ( [GSUBO(BC)] ) * [PF];      !
495 6!      IF BFLUTR <> 0 OR BGUST <> 0 OR BBLAST <> 0 THEN      !
496 7!      [TMP1] := TRANS ( [UGTKF] ) * [GSUBO(BC)];      !
497 7!      CALL TRNSPOSE ( [TMP1], [UGTKA] );      !
498 7!      EN IF;      !
499 6!      ELSE      !
500 6!      IF NOMIT <> 0 THEN      !
501 7!$                                     $!
502 7!$      PERFORM THE STATIC REDUCTION      $!
503 7!$                                     $!
504 7!      PRINT('LOG=( STATIC CONDENSATION )');      !
505 7!$                                     $!
506 7!      CALL FREDUCE ( [KFF], [PF], [PFOA(BC)], , [KOOINV(BC)], ,      !
507 7!      [GSUBO(BC)] [KAA], [PA], [PO], USET(BC) );      !
508 7!$                                     $!
509 7!      IF BMASS <> 0 THEN      !
510 8!$                                     $!
511 8!$      PERFORM GUYAN REDUCTION OF THE MASS MATRIX      $!
512 8!$                                     $!

```


Standard MAPOL Sequence Listing — Continued

```

STAT LEVL
513 8: CALL PARTN ( [MFF], [MOO], , [MOA], [MAABAR], [PFOA(BC)] );!
514 8: [MAA] := [MAABAR] + TRANS([MOA]) * [GSUBO(BC)] + !
515 8: TRANS([GSUBO(BC)]) * [MOA] + !
516 8: TRANS([GSUBO(BC)]) * [ [MOO] * [GSUBO(BC)] ]; !
517 8: IF NRSET <> 0 [IFM(BC)] := [MOO] * [GSUBO(BC)] + [MOA]; !
518 8: ENDIF; !
519 7: IF BFLUTR <> 0 OR BGUST <> 0 OR BBLAST <> 0 THEN !
520 8: CALL ROWPART ( [UGTKF], [UGTKO], [UGTKAB], [PFOA(BC)] ); !
521 8: [TMP1] := TRANS( [UGTKO] ) * [GSUBO(BC)]; !
522 8: CALL TRNSPOSE ( [TMP1], [TMP2] ); !
523 8: [UGTKA] := [UGTKAB] + [TMP2]; !
524 8: ENDIF; !
525 7: ELSE !
526 7:$ $!
527 7:$ NO F-SET REDUCTION $!
528 7:$ $!
529 7: [KAA] := [KFF]; !
530 7: IF BLOAD <> 0 [PA] := [PF]; !
531 7: IF BFLUTR <> 0 OR BGUST <> 0 OR BBLAST <> 0 [UGTKA]:=[UGTKF]; !
532 7: IF BMASS <> 0 [MAA] := [MFF]; !
533 7: ENDIF; !
534 6: ENDIF; !
535 5:$ $!
536 5: IF NRSET <> 0 THEN !
537 6:$ $!
538 6:$ PERFORM THE SUPPORT SET REDUCTION $!
539 6:$ $!
540 6: PRINT("LOG=(' SUPPORT REDUCTION')"); !
541 6: IF NITER = 1 THEN !
542 7: CALL PARTN ( [KAA], [KRR], [KLR], , [KLL], [PARL(BC)] ); !
543 7: CALL SDCOMP ( [KLL], [KLLINV(BC)], USET(BC), SINGLSET ); !
544 7: CALL FBS ( [KLLINV(BC)], [KLR], [D(BC)], -1 ); !
545 7: CALL RBCHECK ( BC, USET(BC), BGPDT(BC), [D(BC)], [KLL], !
546 7: [KRR], [KLR] ); !
547 6: ELSE !
548 7: IF BLOAD <> 0 THEN !
549 8: CALL PARTN ( [KAA], , [KLR], , [KLL], [PARL(BC)] ); !
550 8: CALL SDCOMP ( [KLL], [KLLINV(BC)], USET(BC), SINGLSET ); !
551 8: ENDIF; !
552 7: ENDIF; !
553 6:$ $!
554 6:$ CALCULATE THE REDUCED MASS MATRIX $!
555 6:$ $!
556 6: CALL PARTN ([MAA], [MRRBAR], [MLR], , [MLL], [PARL(BC)]); !
557 6: [IFR(BC)] := [MLL] * [D(BC)] + [MLR]; !
558 6: [MRR(BC)] := [MRRBAR] + TRANS ( [MLR] ) * [D(BC)] + !
559 6: TRANS ( [D(BC)] ) * [IFR(BC)]; !
560 6: [R22] := TRANS ( [D(BC)] ) * [MLR] + [MRRBAR]; !
561 6:$ $!
562 6: IF BLOAD <> 0 THEN !
563 7:$ $!
564 7:$ PROCESS STATICS WITH INERTIA RELIEF $!

```

Standard MAPOL Sequence Listing — Continued

```

STAT LEVL
565 7!$
566 7!
567 7! PRINT(
568 7! "LOG=(' >>>DISCIPLINE: STATICS(INERTIA RELIEF)');
569 7! CALL ROWPART ( {PA}, {PR}, {PLBAR}, {PARL(BC)} );
570 7! {LHS(BC)} := {MRR(BC)};
571 7! {RHS(BC)} := TRANS([D(BC)]) * {PLBAR} + {PR};
572 7! CALL INERTIA ( {LHS(BC)}, {RHS(BC)}, {AR} );
573 7! {AL} := [D(BC)] * {AR};
574 7! CALL ROWMERGE ( {AA}, {AR}, {AL}, {PARL(BC)} );
575 7! {RHS(BC)} := {PLBAR} - {IFR(BC)} * {AR};
576 7! CALL FBS ( {KLLINV(BC)}, {RHS(BC)}, {UL} );
577 7! CALL YSMERGE ( {UA}, , {UL}, {PARL(BC)} );
578 6! ENDF;
579 7! IF BMODES <> 0 THEN
580 7! PRINT("LOG=(' >>>DISCIPLINE: NORMAL MODES')");
581 7! CALL REIG ( NITER, BC, USET(BC), {KAA}, {MAA}, {MRR(BC)},
582 7! [D(BC)], LAMBDA, {PHIA}, {MII}, HSIZE(BC) );
583 7! CALL OFFPMROOT ( NITER, BC, NUMOPTBC, LAMBDA );
584 7! CALL FCEVAL ( NITER, BC, LAMBDA, CONST );
585 6! ENDF;
586 6!$ ELSE
587 6!$ NO SUPPORT SET REDUCTION
588 6!$
589 6! IF BLOAD <> 0 THEN
590 7! PRINT("LOG=(' >>>DISCIPLINE: STATICS')");
591 7! CALL SDCOMP ( {KAA}, {KLLINV(BC)}, USET(BC), SINGASET );
592 7! CALL FBS ( {KLLINV(BC)}, {PA}, {UA} );
593 7! ENDF;
594 6! IF BMODES <> 0 THEN
595 7! PRINT("LOG=(' >>>DISCIPLINE: NORMAL MODES')");
596 7! CALL REIG ( NITER, BC, USET(BC), {KAA}, {MAA}, , , LAMBDA,
597 7! {PHIA}, {MII}, HSIZE(BC) );
598 7! CALL OFFPMROOT ( NITER, BC, NUMOPTBC, LAMBDA );
599 7! CALL FCEVAL ( NITER, BC, LAMBDA, CONST );
600 7! ENDF;
601 6! ENDF;
602 5! ENDF;
603 4! IF BSAERO <> 0 THEN
604 5!$
605 5!$ PERFORM STATIC AEROELASTIC ANALYSES
606 5!$
607 5! PRINT("LOG=(' SAERO INITIALIZATION')");
608 5! CALL TRNSPOSE ( {GSTKF}, {GSKF} );
609 5! LOOP := TRUE;
610 5! SUB := 0;
611 5! WHILE LOOP DO
612 6! SUB := SUB + 1;
613 6! CALL SAERODRV ( BC, SUB, LOOP, MINDEX, SYM, MACH, QDP, 1 );
614 6!$
615 6!$ ADJUST THE KFF MATRIX AND DETERMINE THE RIGID AIR LOADS
616 6!$

```

Standard MAPOL Sequence Listing — Continued

```

STAT LEVEL
617 6! IF SYM = 1 [AICS] := [GTKF]*[TRANS([AICMAT(MINDEX)])]*[GSKF]; !
618 6! IF SYM = -1 [AICS] := [GTKF]*[TRANS([AAICMAT(MINDEX)])]*[GSKF]; !
619 6! [PAF] := (QDP) [GTKF] * [AIRFC(MINDEX)]; !
620 6! [KAFF] := [KFF] - (QDP) [AICS]; !
621 6!$ !
622 6!$ REDUCE THE MATRICES WITH AEROELASTIC CORRECTIONS $!
623 6!$ SAVE THE SUBCASE/BC DEPENDENT DATA FOR SENSITIVITY ANALYSIS $!
624 6!$ !
625 6! IF NGDR <> 0 THEN !
626 7!$ !
627 7!$ PERFORM THE GENERAL DYNAMIC REDUCTION $!
628 7!$ !
629 7! PRINT("LOG=(' SAERO DYNAMIC REDUCTION')"); !
630 7! [MAAA] := TRANS ([GSUBO(BC)]) * [ [MFF] * [GSUBO(BC)] ]; !
631 7! [KAAA] := TRANS ([GSUBO(BC)]) * [ [KAFF] * [GSUBO(BC)] ]; !
632 7! [PAA] := TRANS ([GSUBO(BC)]) * [PAF]; !
633 7! ELSE !
634 7! IF NOMIT <> 0 THEN !
635 8!$ !
636 8!$ PERFORM THE STATIC REDUCTION $!
637 8!$ !
638 8! PRINT("LOG=(' SAERO STATIC CONDENSATION')"); !
639 8!$ !
640 8! IF NITER = 1 AND SUB = 1 AND NRSET <> 0 AND BLOAD = 0 AND !
641 9! BMODES = 0 AND BFLUTR = 0 AND BDYN = 0 THEN !
642 9!$ !
643 9!$ FORM [KAA] ON FIRST PASS SO [D] CAN BE FORMED $!
644 9!$ !
645 9! CALL FREDUCE ([KFF], , [PFOA(BC)], , [KOOINV(BC)], , , !
646 9! [GSUBO(BC)], [KAA], , , USET(BC) ); !
647 9! ENDDIF; !
648 8!$ !
649 8! CALL FREDUCE ( [KAFF], [PAF], [PFOA(BC)], BSAERO, !
650 8! [KOOI(BC,SUB)], [KOOO(BC,SUB)], !
651 8! [KAO(BC,SUB)], [GASUBO(BC,SUB)], [KAAA], !
652 8! [PAA], [POARO(BC,SUB)], USET(BC)); !
653 8!$ !
654 8! IF BMAS <> 0 THEN !
655 9!$ !
656 9!$ PERFORM GUYAN REDUCTION OF THE MASS MATRIX $!
657 9!$ !
658 9! CALL PARTN ( [MFF], [MOO], , [MOA], [MAABAR], !
659 9! [PFOA(BC)] ); !
660 9! [MAAA] := [MAABAR] + TRANS([MOA]) * [GASUBO(BC,SUB)] + !
661 9! TRANS([GASUBO(BC,SUB)]) * [MOA] + !
662 9! TRANS([GASUBO(BC,SUB)]) * [[MOO] * !
663 9! [GASUBO(BC,SUB)]]; !
664 9! IF NRSET <> 0 !
665 10! [IFMA(BC,SUB)] := [MOO]*[GASUBO(BC,SUB)]+[MOA]; !
666 9! ENDDIF; !
667 8! ELSE !
668 8!$ !

```

Standard MAPOL Sequence Listing — Continued

```

STAT LEVL
669 8!$ NO F-SET REDUCTION $!
670 8!$ $!
671 8! IF NITER = 1 AND SUB = 1 AND NRSET <> 0 AND BLOAD = 0 AND $!
672 9! BMODES = 0 AND BFLUTR = 0 AND BDDYN = 0 THEN $!
673 9!$ $!
674 9!$ FORM [KAA] ON FIRST PASS SO [D] CAN BE FORMED $!
675 9!$ $!
676 9! [KAA] := [KFF]; $!
677 9! ENDIF; $!
678 8! [KAAA] := [KAFF]; $!
679 8! [MAAA] := [MFF]; $!
680 8! [PAA] := [PAF]; $!
681 8! ENDIF; $!
682 7! ENDIF; $!
683 6!$ $!
684 6! IF NRSET <> 0 THEN $!
685 7!$ $!
686 7!$ PERFORM THE SUPPORT SET REDUCTION $!
687 7!$ $!
688 7! PRINT("LOG=(' SAERO SUPPORT REDUCTION')"); $!
689 7!$ $!
690 7! IF NITER = 1 AND SUB = 1 AND BLOAD = 0 AND BMODES = 0 AND $!
691 8! BFLUTR = 0 AND BDDYN = 0 THEN $!
692 8!$ $!
693 8!$ [D] WAS NOT COMPUTED FOR NON-SAERO DISCIPLINES SO $!
694 8!$ NEED TO COMPUTE IT NOW $!
695 8!$ $!
696 8! CALL PARTN ( [KAA], [KRR], [KLR], , [KLL], [PARL(BC)] ); $!
697 8! CALL SDCOMP ( [KLL], [KLLINV(BC)], USET(BC), SINGLSET ); $!
698 8! CALL FBS ( [KLLINV(BC)], [KLR], [D(BC)], -1 ); $!
699 8! CALL RBCHECK ( BC, USET(BC), BGPDT(BC), [D(BC)], [KLL], $!
700 8! [KRR], [KLR] ); $!
701 8! ENDIF; $!
702 7!$ $!
703 7!$ CALCULATE THE REDUCED MASS MATRIX $!
704 7!$ $!
705 7! CALL PARTN ([MAAA], [MRRBAR], [MLR], , [MLL], [PARL(BC)]); $!
706 7! [R13(BC,SUB)] := [MLL] * [D(BC)] + [MLR]; $!
707 7! [R33] := [MRRBAR] + TRANS ( [MLR] ) * [D(BC)] + $!
708 7! TRANS ( [D(BC)] ) * [R13(BC,SUB)]; $!
709 7! [R22] := TRANS ( [D(BC)] ) * [MLR] + [MRRBAR]; $!
710 7! CALL TRNSPOSE ( [R13(BC,SUB)], [R21(BC,SUB)] ); $!
711 7!$ $!
712 7!$ PROCESS STEADY AEROELASTIC DISCIPLINE $!
713 7!$ $!
714 7! PRINT("LOG=(' >>>DISCIPLINE: STEADY AERO')"); $!
715 7! CALL PARTN ( [KAAA], [KARR], [R12(BC,SUB)], [KARL], [R11], $!
716 7! [PARL(BC)] ); $!
717 7! [R32(BC,SUB)] := TRANS([D(BC)]) * [R12(BC,SUB)] + [KARR]; $!
718 7! [R31(BC,SUB)] := TRANS([D(BC)]) * [R11] + [KARL]; $!
719 7!$ $!
720 7! CALL DECOMP ( [R11], [RL11(BC,SUB)], [RU11(BC,SUB)] ); $!

```

Standard MAPOL Sequence Listing — Continued

```

STAT LEVL
721 7!$
722 7! CALL ROWPART ( [PAA], [PARBAR], [PAL], [PARL(BC)] );
723 7! CALL GFBS ( [RL11(BC,SUB)], [RU11(BC,SUB)], [PAL],
724 7! [R11PAL(BC,SUB)], -1);
725 7! [PRIGID] := [PARBAR] + TRANS([D(BC)]) * [PAL];
726 7! [P1] := [R21(BC,SUB)] * [R11PAL(BC,SUB)];
727 7! [P2] := [PRIGID] + [R31(BC,SUB)] * [R11PAL(BC,SUB)];
728 7!$
729 7! CALL GFBS ( [RL11(BC,SUB)], [RU11(BC,SUB)], [R12(BC,SUB)],
730 7! [R1112(BC,SUB)], -1);
731 7! CALL GFBS ( [RL11(BC,SUB)], [RU11(BC,SUB)], [R13(BC,SUB)],
732 7! [R1113(BC,SUB)], -1);
733 7! [K11] := [R22] + [R21(BC,SUB)] * [R1112(BC,SUB)];
734 7! [K12(BC,SUB)] := [R21(BC,SUB)] * [R1113(BC,SUB)];
735 7! [K21(BC,SUB)] := [R32(BC,SUB)] +
736 7! [R31(BC,SUB)] * [R1112(BC,SUB)];
737 7! [K22] := [R33] + [R31(BC,SUB)] * [R1113(BC,SUB)];
738 7!$
739 7! CALL DECOMP ( [K11], [KL11(BC,SUB)], [KU11(BC,SUB)] );
740 7! CALL GFBS ( [KL11(BC,SUB)], [KU11(BC,SUB)], [P1],
741 7! [PAR(BC,SUB)] );
742 7! CALL GFBS ( [KL11(BC,SUB)], [KU11(BC,SUB)], [K12(BC,SUB)],
743 7! [K1112(BC,SUB)], -1);
744 7! [LHSA(BC,SUB)] := [K22] + [K21(BC,SUB)] * [K1112(BC,SUB)];
745 7! [RHSA(BC,SUB)] := [P2] - [K21(BC,SUB)] * [PAR(BC,SUB)];
746 7! CALL SAERO ( NITER, BC, MINDEX, SUB, SYM, QDP, STABCF,
747 7! BGPDT(BC), [LHSA(BC,SUB)], [RHSA(BC,SUB)], [AAR],
748 7! [DELTA(SUB)], [PRIGID], [R33],
749 7! CONST, AEFLG(SUB), [AARC], [DELCL]);
750 7! [AAL] := [D(BC)] * [AAR];
751 7! CALL ROWMERGE ( [AAA(SUB)], [AAR], [AAL], [PARL(BC)] );
752 7! [UAR] := [K1112(BC,SUB)] * [AAR] + [PAR(BC,SUB)] *
753 7! [DELTA(SUB)];
754 7! [UAL] := [R1112(BC,SUB)] * [UAR] + [R1113(BC,SUB)] * [AAR]
755 7! - [R11PAL(BC,SUB)] * [DELTA(SUB)];
756 7! CALL ROWMERGE ( [UAA(SUB)], [UAR], [UAL], [PARL(BC)] );
757 7! IF NOMIT <> 0 [PAO(SUB)] := [POARO(BC,SUB)] * [DELTA(SUB)];
758 7! IF AEFLG(SUB) THEN
759 8! [AAL] := [D(BC)] * [AARC];
760 8! CALL ROWMERGE ( [AAAC(SUB)], [AARC], [AAL], [PARL(BC)] );
761 8! [UAR] := [K1112(BC,SUB)] * [AARC] + [PAR(BC,SUB)] *
762 8! [DELCL];
763 8! [UAL] := [R1112(BC,SUB)] * [UAR] +
764 8! [R1113(BC,SUB)] * [AARC] -
765 8! [R11PAL(BC,SUB)] * [DELCL];
766 8! CALL ROWMERGE ( [UAAC(SUB)], [UAR], [UAL], [PARL(BC)] );
767 8! IF NOMIT <> 0 [PAOC(SUB)] := [POARO(BC,SUB)] * [DELCL];
768 8! ENDIF;
769 7! ELSE
770 7!$
771 7!$ NO SUPPORT SET REDUCTION
772 7!$ PROCESS STEADY AEROELASTIC DISCIPLINE

```

Standard MAPOL Sequence Listing — Continued

```

STAT  LEVEL
773      7!$
774      7!      PRINT("LOG=('          >>>DISCIPLINE: STEADY AERO')");      $!
775      7!$
776      7!      ENDIF;      !
777      6!      ENDDO;      !
778      5!      ENDIF;      !
779      4!$      $!
780      4!$      PERFORM ANY DYNAMIC ANALYSES -- NOTE THAT THESE ARE INDEPENDENT      $!
781      4!$      OF THE SUPPORT SET      $!
782      4!$      $!
783      5!      IF BDYN <> 0 THEN      !
784      4!      IF BFLUTR <> 0 THEN      !
785      6!      PRINT("LOG=('          >>>DISCIPLINE: FLUTTER')");      !
786      6!      SUB := 0;      !
787      6!      LOOP := TRUE;      !
788      6!      WHILE LOOP DO      !
789      7!      SUB := SUB + 1;      !
790      7!      CALL FLUTDRV ( BC, SUB, LOOP );      !
791      7!      CALL FLUTQHHL ( NITER, BC, SUB, ESIZE(BC), PSIZE(BC), [QKKL],      !
792      7!      [UGTKA], [PHIA], USET(BC),      !
793      7!      [TMN(BC)], [GSUBO(BC)], NGDR, AECOMPU, GEOMUA,      !
794      7!      [PHIKH], [QHHLFL(BC,SUB)], OAGRDDSP );      !
795      7!      CALL FLUTDMA ( NITER, BC, SUB, ESIZE(BC), PSIZE(BC),      !
796      7!      BGPDT(BC), USET(BC), [MAA], [KAA], [TMN(BC)],      !
797      7!      [GSUBO(BC)], NGDR, LAMBDA, [PHIA],      !
798      7!      [MHHFL(BC,SUB)], [BHHFL(BC,SUB)], [KHHFL(BC,SUB)] );      !
799      7!      CALL FLUTTRAN ( NITER, BC, SUB, [QHHLFL(BC,SUB)], LAMBDA,      !
800      7!      HSIZE(BC), ESIZE(BC), [MHHFL(BC,SUB)],      !
801      7!      [BHHFL(BC,SUB)], [KHHFL(BC,SUB)],      !
802      7!      CLAMBDA, CONST );      !
803      7!      ENDDO;      !
804      6!      ENDIF;      !
805      5!$      $!
806      5!      IF BDRSP <> 0 THEN      !
807      6!      IF BMTR <> 0 OR BDTR <> 0 THEN      !
808      7!      PRINT("LOG=('          >>>DISCIPLINE: TRANSIENT RESPONSE')");      !
809      7!      ENDIF;      !
810      6!      IF BMFR <> 0 OR BDFR <> 0 THEN      !
811      7!      PRINT("LOG=('          >>>DISCIPLINE: FREQUENCY RESPONSE')");      !
812      7!      ENDIF;      !
813      6!      CALL QHHLGEN (BC, ESIZE(BC), [QKKL], [QKJL], [UGTKA], [PHIA],      !
814      6!      [PHIKH], [QHHL], [QHJL]);      !
815      6!      CALL DMA ( NITER, BC, ESIZE(BC), PSIZE(BC), BGPDT(BC), USET(BC),      !
816      6!      [MAA], [KAA], [TMN(BC)], [GSUBO(BC)], NGDR,      !
817      6!      LAMBDA, [PHIA], [MDD], [BDD], [KDDT], [KDDF],      !
818      6!      [MHH], [BHH], [KHHT], [KHHF] );      !
819      6!      CALL DYNLOAD ( NITER, BC, GSIZE, ESIZE(BC), PSIZE(BC), SMPLOD,      !
820      6!      BGPDT(BC), USET(BC), [TMN(BC)], [GSUBO(BC)],      !
821      6!      NGDR, [PHIA], [QHJL], [PDT], [PDF],      !
822      6!      [PTGLOAD], [PTHLOAD], [PFGLOAD], [PFHLOAD] );      !
823      6!      CALL DYNRSP (BC, ESIZE(BC), [MDD], [BDD], [KDDT], [KDDF],      !
824      6!      [MHH], [BHH], [KHHT], [KHHF], [PDT], [PDF],      !

```

Standard MAPOL Sequence Listing — Continued

```

STAT LEVL
825 6! [QHHL], [UTRANA], [UFREQA], [UTRANI], [UFREQI], !
826 6! [UTRANE], [UFREQE] ); !
827 6! IF BMTR <> 0 [UTRANA] := [PHIA] * [UTRANI]; !
828 6! IF BMFR <> 0 [UFREQA] := [PHIA] * [UFREQI]; !
829 6! ENDIF; !
830 5! ENDIF; !
831 4! IF BBLAST <> 0 THEN !
832 5! PRINT("LOG=(' >>>DISCIPLINE: BLAST')"); !
833 5! CALL BLASTFIT ( BC, [QJL], [MATTR], [MATSS], BQDP, [BFRC], !
834 5! [DWNWSH], HSIZE(BC), [ID2], [MPART], [UGTKA], !
835 5! [BLGTJA], [BLSTJA] ); !
836 5! CALL COLPART ( [PHIA], , [PHIE], [MPART] ); !
837 5! CALL ROWMERGE ( [PHIR], [ID2], [D(BC)], [PARL(BC)] ); !
838 5! CALL COLMERGE ( [PHIB], [PHIR], [PHIE], [MPART] ); !
839 5! [GENM] := TRANS( [PHIB] ) * [ [MAA] * [PHIB] ]; !
840 5! [GENK] := TRANS( [PHIB] ) * [ [KAA] * [PHIB] ]; !
841 5! [DTSLP] := TRANS ( [BLSTJA] ) * [PHIB]; !
842 5! [FTF] := TRANS ( [PHIB] ) * [BLGTJA]; !
843 5! [GENF] := (BQDP) [FTF] * [BFRC]; !
844 5! [GENFA] := (BQDP) [FTF] * [MATSS]; !
845 5! [GENQ] := [GENFA] * [DTSLP]; !
846 5! [GENQL] := (BQDP) [FTF] * [MATTR]; !
847 5! CALL PARTN ( [GENQ], [QRR] , , [QRE], [QEE], [MPART] ); !
848 5! CALL PARTN ( [GENK], , , [KEE], [MPART] ); !
849 5! [KEQE] := [QEE] + [KEE]; !
850 5! CALL DECOMP ( [KEQE], [LKQ], [UKQ] ); !
851 5! CALL ROWPART ( [GENF], [GFR], [GFE], [MPART] ); !
852 5! CALL GFBS ( [LKQ], [UKQ], [GFE], [BTEM] ); !
853 5! [DELM] := -[QRE] * [BTEM] + [GFR]; !
854 5! CALL BLASTRIM ( BC, [DELM], [MRR(BC)], [URDB], [DELB] ); !
855 5! [ELAS] := [BTEM] * [DELB]; !
856 5! [SLPMOD] := TRANS ( [BLSTJA] ) * [PHIE]; !
857 5! CALL BLASTDRV ( BC, [GENM], [GENK], [GENFA], [GENQL], [DELB], !
858 5! [URDB], [DWNWSH], [SLPMOD], [ELAS], [UBLASTI] ); !
859 5! ENDIF; !
860 4!$ $!
861 4!$ BEGIN THE DATA RECOVERY OPERATIONS $!
862 4!$ $!
863 4! PRINT("LOG=(' DATA RECOVERY')"); !
864 4! IF NUMOPTBC > 1 CALL NULLMAT ([UF], [AF], [PHIF], [UTRANF], [UFREQF]); !
865 4! IF NGDR <> 0 THEN !
866 5!$ $!
867 5!$ DATA RECOVERY WITH GDR $!
868 5!$ APPEND THE GDR-GENERATED DOFS TO THE F-SET $!
869 5!$ $!
870 5! PRINT("LOG=(' DYNAMIC REDUCTION RECOVERY')"); !
871 5! IF BLOAD <> 0 THEN !
872 6! [UFGDR] := [GSUBO(BC)] * [UA]; !
873 6! CALL ROWPART ( [UA], [UJK], , [PAJK] ); !
874 6! CALL ROWMERGE ( [UF], [UJK], [UFGDR], [PFJK] ); !
875 6! IF NRSET <> 0 THEN !
876 7! [AFGDR] := [GSUBO(BC)] * [AA]; !

```

Standard MAPOL Sequence Listing — Continued

```

STAT LEVL
877 7! CALL ROWPART ( [AA], [UJK], , [PAJK] );
878 7! CALL ROWMERGE ( [AF], [UJK], [AFGDR], [PFJK] );
879 7! ENDF;
880 6! ENDF;
881 5! IF BSAERO <> 0 THEN
882 6! FOR S = 1 TO SUB DO
883 7! [UFGDR] := [GSUBO(BC)] * [UAA(S)];
884 7! CALL ROWPART ( [UAA(S)], [UJK], , [PAJK] );
885 7! CALL ROWMERGE ( [UAFTMP], [UJK], [UFGDR], [PFJK] );
886 7!$
887 7!$ MERGE THE CURRENT SUBCASE DEPENDENT RESULTS INTO A SINGLE
888 7!$ MATRIX OF RESPONSE QUANTITIES FOR FURTHER RECOVERY
889 7!$
890 7! CALL SAEROMRG ( BC, S, [UAF], [UAFTMP] );
891 7! IF NRSET <> 0 THEN
892 8! [AFGDR] := [GSUBO(BC)] * [AAA(S)];
893 8! CALL ROWPART ( [AAA(S)], [UJK], , [PAJK] );
894 8! CALL ROWMERGE ( [AAFTMP], [UJK], [AFGDR], [PFJK] );
895 8! CALL SAEROMRG ( BC, S, [AAF], [AAFTMP] );
896 8! ENDF;
897 7! IF AEFLG(S) THEN
898 8! [UFGDR] := [GSUBO(BC)] * [UAAC(S)];
899 8! CALL ROWPART ( [UAAC(S)], [UJK], , [PAJK] );
900 8! CALL ROWMERGE ( [UAFC(S)], [UJK], [UFGDR], [PFJK] );
901 8! [AFGDR] := [GSUBO(BC)] * [AAAC(S)];
902 8! CALL ROWPART ( [AAAC(S)], [UJK], , [PAJK] );
903 8! CALL ROWMERGE ( [AAFC(S)], [UJK], [AFGDR], [PFJK] );
904 8! ENDF;
905 7! ENDDO;
906 6! ENDF;
907 5! IF BMODES <> 0 THEN
908 6! [UFGDR] := [GSUBO(BC)] * [PHIA];
909 6! CALL ROWPART ( [PHIA], [UJK], , [PAJK] );
910 6! CALL ROWMERGE ( [PHIF], [UJK], [UFGDR], [PFJK] );
911 6! ENDF;
912 5! IF BDTR <> 0 OR BMTR <> 0 THEN
913 6! [UFGDR] := [GSUBO(BC)] * [UTRANA];
914 6! CALL ROWPART ( [UTRANA], [UJK], , [PAJK] );
915 6! CALL ROWMERGE ( [UTRANF], [UJK], [UFGDR], [PFJK] );
916 6! ENDF;
917 5! IF BDFR <> 0 OR BMFR <> 0 THEN
918 6! [UFGDR] := [GSUBO(BC)] * [UFREQA];
919 6! CALL ROWPART ( [UFREQA], [UJK], , [PAJK] );
920 6! CALL ROWMERGE ( [UFREQF], [UJK], [UFGDR], [PFJK] );
921 6! ENDF;
922 5! ELSE
923 5! IF NOMIT <> 0 THEN
924 6!$
925 6!$ DATA RECOVERY WITH STATIC CONDENSATION
926 6!$
927 6! PRINT("LOG= '          STATIC CONDENSATION RECOVERY'");
928 6! IF BLOAD <> 0 THEN

```


Standard MAPOL Sequence Listing — Continued

```

STAT LEVL
929 7! CALL RECOVA ( [UA], [PO], [GSUBO(BC)], NRSET, [AA], !
930 7! [IFM(BC)], , [KOOINV(BC)], [PFOA(BC)], [UF] ); !
931 7! IF NRSET <> 0 CALL RECOVA ( [AA], , [GSUBO(BC)], , , !
932 8! [PFOA(BC)], [AF] ); !
933 7! ENDIF; !
934 6! IF BSAERO <> 0 THEN !
935 7! FOR S = 1 TO SUB DO !
936 8! CALL RECOVA ( [UAA(S)], [PAO(S)], [GASUBO(BC,S)], !
937 8! NRSET, [AAA(S)], [IFMA(BC,S)], BSAERO, !
938 8! [KOOL(BC,S)], [KOOU(BC,S)], !
939 8! [PFOA(BC)], [UAFTMP] ); !
940 8!$ $!
941 8!$ MERGE THE CURRENT SUBCASE DEPENDENT RESULTS INTO A SINGLE $!
942 8!$ MATRIX OF RESPONSE QUANTITIES FOR FURTHER RECOVERY $!
943 8!$ $!
944 8! CALL SAEROMRG ( BC, S, [UAF], [UAFTMP] ); !
945 8! IF NRSET <> 0 THEN !
946 9! CALL RECOVA ( [AAA(S)], [GASUBO(BC,S)], , , !
947 9! [PFOA(BC)], [AAFTMP] ); !
948 9! CALL SAEROMRG ( BC, S, [AAF], [AAFTMP] ); !
949 9! ENDIF; !
950 8! IF AEFLG(S) THEN !
951 9! CALL RECOVA ( [UAAC(S)], [PAOC(S)], [GASUBO(BC,S)], !
952 9! NRSET, [AAAC(S)], [IFMA(BC,S)], BSAERO, !
953 9! [KOOL(BC,S)], [KOOU(BC,S)], !
954 9! [PFOA(BC)], [UAFC(S)] ); !
955 9! CALL RECOVA ( [AAAC(S)], [GASUBO(BC,S)], , , !
956 9! [PFOA(BC)], [AAFC(S)] ); !
957 9! ENDIF; !
958 8! ENDDO; !
959 7! ENDIF; !
960 6! IF BMODES <> 0 THEN !
961 7! [PHIO] := [GSUBO(BC)] * [PHIA]; !
962 7! CALL ROWMERGE ( [PHIF], [PHIO], [PHIA], [PFOA(BC)] ); !
963 7! ENDIF; !
964 6! IF BDTR <> 0 OR BMTR <> 0 THEN !
965 7! CALL RECOVA ( [UTRANA], , [GSUBO(BC)], , , !
966 7! [PFOA(BC)], [UTRANF] ); !
967 7! ENDIF; !
968 6! IF BDFR <> 0 OR BMFR <> 0 THEN !
969 7! CALL RECOVA ( [UFREQA], , [GSUBO(BC)], , , !
970 7! [PFOA(BC)], [UFREQF] ); !
971 7! ENDIF; !
972 6! ELSE !
973 6!$ $!
974 6!$ DATA RECOVERY WITHOUT F-SET REDUCTION $!
975 6!$ $!
976 6! IF BLOAD <> 0 THEN !
977 7! [UF] := [UA]; !
978 7! IF NRSET <> 0 [AF] := [AA]; !
979 7! ENDIF; !
980 6! IF BSAERO <> 0 THEN !

```

Standard MAPOL Sequence Listing — Continued

```

STAT LEVL
981 7!          FOR S = 1 TO SUB DO                      !
982 8!$          !                                       $!
983 8!$          MERGE THE CURRENT SUBCASE DEPENDENT RESULTS INTO A SINGLE $!
984 8!$          MATRIX OF RESPONSE QUANTITIES FOR FURTHER RECOVERY $!
985 8!$          !                                       $!
986 8!          CALL SAEROMRG ( BC, S, [UAF], [UAA(S)] );    !
987 8!          IF NRSET <> 0 CALL SAEROMRG ( BC, S, [AAF], [AAA(S)] );    !
988 8!          IF AEFLG(S) THEN                          !
989 9!              [UAPC(S)] := [UAPC(S)];                !
990 9!              [AAFC(S)] := [AAFC(S)];                !
991 9!          ENDIF;                                     !
992 8!          ENDDO;                                     !
993 7!          ENDF;                                     !
994 6!          IF BMODES <> 0 [PHIF] := [PHIA];            !
995 6!          IF BDTR <> 0 OR BMTR <> 0 [UTRANF] := [UTRANA];    !
996 6!          IF BDFR <> 0 OR BMFR <> 0 [UFREQF] := [UFREQA];    !
997 6!          ENDF;                                     !
998 5!          ENDF;                                     !
999 4!$          !                                       $!
1000 4!          IF NUMOPTBC > 1 CALL NULLMAT ( [UN], [AN], [PHIN] );    !
1001 4!          IF NSPC <> 0 THEN                          !
1002 5!$          !                                       $!
1003 5!$          DATA RECOVERY WITH SPC-REDUCTION        !
1004 5!$          !                                       $!
1005 5!          PRINT("LOG=('          SPC RECOVERY')");    !
1006 5!          IF BLOAD <> 0 THEN                          !
1007 6!              CALL YSMERGE ( [UN], [YS(BC)], [UF], [PNSF(BC)] );    !
1008 6!              CALL OFFSPCF ( NITER, BC, 1, 1, GSIZE, ESIZE(BC), NGDR,    !
1009 6!                  [KFS], [KSS], [UF], [YS(BC)], [PS],    !
1010 6!                  [PNSF(BC)], [PGMN(BC)], [PFJK], , , ,    !
1011 6!                  BGPDT(BC), OGRIDLOD );                !
1012 6!              IF NRSET <> 0 CALL YSMERGE ( [AN], , [AF], [PNSF(BC)] );    !
1013 6!          ENDF;                                     !
1014 5!          IF BSAERO <> 0 THEN                          !
1015 6!              CALL YSMERGE ( [UAN], [YS(BC)], [UAF], [PNSF(BC)] );    !
1016 6!              IF NRSET <> 0 CALL YSMERGE ( [AAN], , [AAF], [PNSF(BC)] );    !
1017 6!              FOR S = 1 TO SUB DO                      !
1018 7!                  IF AEFLG(S) THEN                      !
1019 8!                      CALL YSMERGE ([UANC(S)], [YS(BC)], [UAPC(S)], [PNSF(BC)]);    !
1020 8!                      CALL YSMERGE ([AANC(S)], , [AAFC(S)], [PNSF(BC)]);    !
1021 8!                  ENDF;                                !
1022 7!              ENDDO;                                  !
1023 6!          ENDF;                                     !
1024 5!          IF BMODES <> 0 THEN                          !
1025 6!              CALL YSMERGE ( [PHIN], [YS(BC)], [PHIF],    !
1026 6!                  [PNSF(BC)] );                          !
1027 6!              IF DMODES <> 0 CALL OFFSPCF ( NITER, BC, 2, 1, GSIZE,    !
1028 7!                  ESIZE(BC), NGDR,                      !
1029 7!                  [KFS], , [PHIF], , , ,                !
1030 7!                  [PNSF(BC)], [PGMN(BC)], [PFJK],    !
1031 7!                  , , , BGPDT(BC), OGRIDLOD );          !
1032 6!          ENDF;                                     !

```

Standard MAPOL Sequence Listing — Continued

```

STAT LEVL
1033 5!      IF BDTR <> 0 OR BMTR <> 0                      !
1034 6!      CALL YSMERGE ( [UTRANN], [YS(BC)], [UTRANF],    !
1035 6!      [PNSF(BC)], BDTR );                          !
1036 5!      IF BDFR <> 0 OR BMFR <> 0                      !
1037 6!      CALL YSMERGE ( [UFREQN], [YS(BC)], [UFREQF],    !
1038 6!      [PNSF(BC)], BDFR );                          !
1039 5!      IF BBLAST <> 0 THEN                            !
1040 6!      [UBLASTF] := [PHIF]*[UBLASTI];                !
1041 6!      CALL OPPSPCF ( NITER, BC, 8, 1, GSIZE, ESIZE(BC), NGDR, !
1042 6!      [KFS], , [UBLASTF], , , [PNSF(BC)], [PGMN(BC)], !
1043 6!      [PFJK], , , BGPDT(BC), OGRIDLOD );            !
1044 6!      ENDIF;                                         !
1045 5!      ELSE                                          !
1046 5!$      $!                                          !
1047 5!$      DATA RECOVERY WITHOUT SPC-REDUCTION          !
1048 5!$      $!                                          !
1049 5!      IF BLOAD <> 0 THEN                            !
1050 6!      [UN] := [UF];                                  !
1051 6!      IF NRSET <> 0 [AN] := [AF];                    !
1052 6!      ENDIF;                                         !
1053 5!      IF BSAERO <> 0 THEN                            !
1054 6!      [UAN] := [UAF];                                  !
1055 6!      IF NRSET <> 0 [AAN] := [AAF];                  !
1056 6!      FOR S = 1 TO SUB DO                          !
1057 7!      IF AEFLG(S) THEN                              !
1058 8!      [UANC(S)] := [UAFC(S)];                        !
1059 8!      [AANC(S)] := [AAFC(S)];                      !
1060 8!      ENDIF;                                         !
1061 7!      ENDDO;                                         !
1062 6!      ENDIF;                                         !
1063 5!      IF BMODES <> 0 [PHIN] := [PHIF];              !
1064 5!      IF BDTR <> 0 OR BMTR <> 0 [UTRANN] := [UTRANA];  !
1065 5!      IF BDFR <> 0 OR BMFR <> 0 [UFREQN] := [UFREQA]; !
1066 5!      ENDIF;                                         !
1067 4!$      $!                                          !
1068 4!      IF NUMOPTBC > 1 CALL NULLMAT ( [UG(BC)], [AG(BC)], [UAG(BC)], !
1069 5!      [AAG(BC)], [PHIG(BC)] );                      !
1070 4!$      $!                                          !
1071 4!      IF NMPC <> 0 THEN                            !
1072 5!$      $!                                          !
1073 5!$      DATA RECOVERY WITH MPC-REDUCTION          !
1074 5!$      $!                                          !
1075 5!      PRINT("LOG=( '      MPC RECOVERY' )");        !
1076 5!      IF BLOAD <> 0 THEN                            !
1077 6!      [UM] := [TMN(BC)] * [UN];                      !
1078 6!      CALL ROWMERGE ( [UG(BC)], [UM], [UN], [PGMN(BC)] ); !
1079 6!      IF NRSET <> 0 THEN                            !
1080 7!      [UM] := [TMN(BC)] * [AN];                      !
1081 7!      CALL ROWMERGE ( [AG(BC)], [UM], [AN], [PGMN(BC)] ); !
1082 7!      ENDIF;                                         !
1083 6!      ENDIF;                                         !
1084 5!      IF BSAERO <> 0 THEN                            !

```

Standard MAPOL Sequence Listing — Continued

```

STAT LEVL
1085 6:      [UM] := [TMN(BC)] * [UAN];
1086 6:      CALL ROWMERGE ( [UAG(BC)], [UM], [UAN], [PGMN(BC)] );
1087 6:      IF NRSET <> 0 THEN
1088 7:          [UM] := [TMN(BC)] * [AAN];
1089 7:          CALL ROWMERGE ( [AAG(BC)], [UM], [AAN], [PGMN(BC)] );
1090 7:      ENDIF;
1091 6:      FOR S = 1 TO SUB DO
1092 7:          IF AEFLG(S) THEN
1093 8:              [UM] := [TMN(BC)] * [UANC(S)];
1094 8:              CALL ROWMERGE ([UAG(BC,S)], [UM], [UANC(S)], [PGMN(BC)]);
1095 8:              [UM] := [TMN(BC)] * [AANC(S)];
1096 8:              CALL ROWMERGE ([AAG(BC,S)], [UM], [AANC(S)], [PGMN(BC)]);
1097 8:          ENDIF;
1098 7:      ENDDO;
1099 6:      ENDIF;
1100 5:      IF BMODES <> 0 THEN
1101 6:          [UM] := [TMN(BC)] * [PHIN];
1102 6:          CALL ROWMERGE ( [PHIG(BC)], [UM], [PHIN], [PGMN(BC)] );
1103 6:      ENDIF;
1104 5:      IF BDTR <> 0 OR BMTR <> 0 THEN
1105 6:          [UM] := [TMN(BC)] * [UTRANN];
1106 6:          CALL ROWMERGE ( [UTRANG], [UM], [UTRANN], [PGMN(BC)] );
1107 6:      ENDIF;
1108 5:      IF BDFR <> 0 OR BMFR <> 0 THEN
1109 6:          [UM] := [TMN(BC)] * [UFREQN];
1110 6:          CALL ROWMERGE ( [UFREQG], [UM], [UFREQN], [PGMN(BC)] );
1111 6:      ENDIF;
1112 5:      ELSE
1113 5:  $
1114 5:  $
1115 5:  $
1116 5:      IF BLOAD <> 0 THEN
1117 6:          [UG(BC)] := [UN];
1118 6:          IF NRSET <> 0 [AG(BC)] := [AN];
1119 6:      ENDIF;
1120 5:      IF BSAERO <> 0 THEN
1121 6:          [UAG(BC)] := [UAN];
1122 6:          IF NRSET <> 0 [AAG(BC)] := [AAN];
1123 6:          FOR S = 1 TO SUB DO
1124 7:              IF AEFLG(S) THEN
1125 8:                  [UAGC(BC,S)] := [UANC(S)];
1126 8:                  [AAGC(BC,S)] := [AANC(S)];
1127 8:              ENDIF;
1128 7:          ENDDO;
1129 6:      ENDIF;
1130 5:      IF BMODES <> 0 [PHIG(BC)] := [PHIN];
1131 5:      IF BDTR <> 0 OR BMTR <> 0 [UTRANG] := [UTRANN];
1132 5:      IF BDFR <> 0 OR BMFR <> 0 [UFREQG] := [UFREQN];
1133 5:      ENDIF;
1134 4:  $
1135 4:  $
1136 4:  $
DATA RECOVERY WITHOUT MPC-REDUCTION
RECOVER PHYSICAL BLAST DISCIPLINE DISPLACEMENTS

```

Standard MAPOL Sequence Listing — Continued

```

STAT LEVL
1137 4! IF BBLAST <> 0 [UBLASTG] := [PHIG(BC)] * [UBLASTI]; !
1138 4!$ !
1139 4!$ PERFORM CONSTRAINT EVALUATION FOR STATIC DISCIPLINES $!
1140 4!$ !
1141 4! PRINT("LOG=(' CONSTRAINT EVALUATION')"); !
1142 4! IF BLOAD <> 0 THEN !
1143 5! CALL DCEVAL ( NITER, BC, [UG(BC)], CONST ); !
1144 5! CALL SCEVAL ( NITER, BC, [UG(BC)], [SMAT], TREF, [GLBSIG], CONST ); !
1145 5! ENDIF; !
1146 4! IF BSAERO <> 0 THEN !
1147 5! CALL DCEVAL ( NITER, BC, [UAG(BC)], CONST, BSAERO ); !
1148 5! CALL SCEVAL ( NITER, BC, [UAG(BC)], [SMAT], TREF, [GLBSIG], CONST, !
1149 5! BSAERO ); !
1150 5! ENDIF; !
1151 4!$ !
1152 4!$ HANDLE OUTPUT REQUESTS $!
1153 4!$ !
1154 4! PRINT("LOG=(' OUTPUT PROCESSING')"); !
1155 4! IF BSAERO <> 0 THEN !
1156 5!$ !
1157 5!$ RECOVER STATIC AEROELASTIC LOADS DATA $!
1158 5!$ !
1159 5! LOOP := TRUE; !
1160 5! SUB := 0; !
1161 5! WHILE LOOP DO !
1162 6! SUB := SUB + 1; !
1163 6! CALL SAERODRV (BC, SUB, LOOP, MINDEX, SYM, MACH, QDP ); !
1164 6!$ !
1165 6!$ CALL THE TRIMMED LOADS COMPUTATION WITH PROPER MATRICES $!
1166 6!$ !
1167 6! IF SYM = 1 THEN !
1168 7! CALL OFFPALOAD ( NITER, BC, MINDEX, SUB, GSIZE, BGPDT(BC), !
1169 7! [GTKG], [GSTKG], QDP, [AIRFRM(MINDEX)], !
1170 7! [DELTA(SUB)], [AICMAT(MINDEX)], !
1171 7! [UAG(BC)], [MGG], [AAG(BC)], [KFS], !
1172 7! [KSS], [UAF], [YS(BC)], [PNSF(BC)], !
1173 7! [PGMN(BC)], [PFJK], NGDR, USET(BC), !
1174 7! OGRIDLOD ); !
1175 7! ELSE !
1176 7! IF SYM = -1 THEN !
1177 8! CALL OFFPALOAD ( NITER, BC, MINDEX, SUB, GSIZE, BGPDT(BC), !
1178 8! [GTKG], [GSTKG], QDP, [AIRFRM(MINDEX)], !
1179 8! [DELTA(SUB)], [AICMAT(MINDEX)], !
1180 8! [UAG(BC)], [MGG], [AAG(BC)], [KFS], !
1181 8! [KSS], [UAF], [YS(BC)], [PNSF(BC)], !
1182 8! [PGMN(BC)], [PFJK], NGDR, USET(BC), !
1183 8! OGRIDLOD ); !
1184 8! ENDIF; !
1185 7! ENDIF; !
1186 6!$ !
1187 6!$ CALL TO COMPUTE THE TRIMMED LOADS/DISPLACEMENTS ON THE $!
1188 6!$ AERODYNAMIC MODEL $!

```

Standard MAPOL Sequence Listing — Continued

```

STAT LEVL
1189 6!$
1190 6!
1191 7! IF SYM = 1 THEN
1192 7! CALL OFPAEROM ( NITER, BC, MINDEX, SUB, GSIZE, GEOMSA,
1193 7! [GTKG], [GSTKG], QDP, [AIRFRC(MINDEX)],
1194 7! [DELTA(SUB)], [AICMAT(MINDEX)],
1195 7! [UAG(BC)], OAGRDLOD, OAGRDDSP );
1196 7! ELSE
1197 8! IF SYM = -1 THEN
1198 8! CALL OFPAEROM ( NITER, BC, MINDEX, SUB, GSIZE, GEOMSA,
1199 8! [GTKG], [GSTKG], QDP, [AIRFRC(MINDEX)],
1200 8! [DELTA(SUB)], [AICMAT(MINDEX)],
1201 8! [UAG(BC)], OAGRDLOD, OAGRDDSP );
1202 7! ENDIF;
1203 6! ENDDO;
1204 5! ENDIF;
1205 4! IF BDRSP <> 0 THEN
1206 5! CALL OFPDLOAD ( NITER, BC, BGPDT(BC), PSIZE(BC), ESIZE(BC),
1207 5! [PHIG(BC)], [PTGLOAD], [PTHLOAD], [PFGLOAD],
1208 5! [PFHLOAD], OGRIDLOD );
1209 5! IF BDTR <> 0 OR BMTR <> 0
1210 6! CALL OFPSPCF ( NITER, BC, 5, 1, GSIZE, ESIZE(BC),
1211 6! NGDR, [KFS], , [UTRANF], ,
1212 6! [PNSF(BC)], [PGMN(BC)], [PFJK],
1213 6! [PHIG(BC)], [PTGLOAD], [PTHLOAD],
1214 6! BGPDT(BC), OGRIDLOD );
1215 5! IF BDFR <> 0 OR BMFR <> 0
1216 6! CALL OFPSPCF ( NITER, BC, 6, 2, GSIZE, ESIZE(BC),
1217 6! NGDR, [KFS], , [UFREQF], ,
1218 6! [PNSF(BC)], [PGMN(BC)], [PFJK],
1219 6! [PHIG(BC)], [PFGLOAD], [PFHLOAD],
1220 6! BGPDT(BC), OGRIDLOD );
1221 5! ENDIF;
1222 4! CALL OFPLOAD ( NUMOPTBC, BC, NITER, GSIZE, BGPDT(BC), PSIZE(BC),
1223 4! [PG] );
1224 4! CALL OFPDISP ( NUMOPTBC, BC, NITER, GSIZE, BGPDT(BC), ESIZE(BC),
1225 4! PSIZE(BC), OGRIDDSP, [UG(BC)], [AG(BC)], [UAG(BC)],
1226 4! [AAG(BC)], [UBLASTG], , [UTRANG], [UTRANE], [UFREQG],
1227 4! [UFREQE], LAMBDA, [PHIG(BC)] );
1228 4! CALL EDR ( NUMOPTBC, BC, NITER, NDV, GSIZE, EOSUMMRY, EODISC,
1229 4! GLBDES, LOCLVAR, [PTRANS],
1230 4! [UG(BC)], [UAG(BC)], , [UTRANG], [UFREQG], [PHIG(BC)] );
1231 4! CALL OFPEDR ( BC, HSIZE(BC), NITER );
1232 4! ENDDO;
1233 3!$
1234 3!$ SELECT ACTIVE CONSTRAINTS
1235 3!$
1236 3! PRINT("LOG=(' SENSITIVITY ANALYSIS')");
1237 3! CALL ACTCON ( NITER, MAXITER, NRFAC, NDV, GLBDES, LOCLVAR, [PTRANS],
1238 3! EPS, APPCNVRG, GLBCNVRG,
1239 3! CTL, CTLMIN, CONST, [AMAT], DESHIST, PFLAG, OLOCALDV );
1240 3! CALL DESPUNCH ( NITER, PFLAG, OLOCALDV );

```

Standard MAPOL Sequence Listing — Continued

```

STAT LEVL
1241 3:$
1242 3: IF GLBCNVRG OR NITER > MAXITER THEN $!
1243 4:$
1244 4:$ LAST ITERATION OUTPUT $!
1245 4:$
1246 4: FOR BC = 1 TO NUMOPTBC DO $!
1247 5: CALL CFFMROOT ( NITER, BC, NUMOPTBC, LAMBDA, 1 ); !
1248 5: CALL OFPDISP ( NUMOPTBC, BC, NITER, GSIZE, BGPDT(BC), ESIZE(BC), !
1249 5: PSIZE(BC), OGRIDDSP,.....,LAMBDA,,, 1 ); !
1250 5: CALL OFPEDR ( BC, HSIZE(BC), NITER, 1 ); !
1251 5: ENDDO; !
1252 4: ENDF; !
1253 3:$
1254 3: IF NOT GLBCNVRG AND NITER <= MAXITER THEN $!
1255 4:$
1256 4:$ USE APPROPRIATE RESIZING METHOD $!
1257 4:$
1258 4: IF NITER >= FSDS AND NITER <= FSDE THEN !
1259 5: CALL FSD ( NDV, NITER, FSDS, FSDE, MPS, OCS, ALPHA, !
1260 5: CNVRGLIM, GLBDES, LOCLVAR, [PTRANS], CONST, !
1261 5: APPCNVRG, CTL, CTLMIN, DESHIST ); !
1262 5: ENDF; !
1263 4:$
1264 4: IF ( NITER >= MPS AND NITER <= MPE ) OR !
1265 5: ( NITER >= OCS AND NITER <= OCE ) THEN !
1266 5:$
1267 5:$ USE MATHEMATICAL PROGRAMMING OR OC METHODS $!
1268 5:$
1269 5:$ OBTAIN THE SENSITIVITIES OF THE CONSTRAINTS WRT THE $!
1270 5:$ DESIGN VARIABLES $!
1271 5:$
1272 5: CALL MAKDFV ( NITER, NDV, [PMINT], [PMAXT], CONST, [AMAT] ); !
1273 5: CALL LAMINSNS ( NITER, NDV, G'DES, LOCLVAR, [PTRANS], CONST, !
1274 5: [AMAT] ); !
1275 5:$
1276 5:$.....$!
1277 5:$ SENSITIVITY EVALUATION FOR BOUNDARY CONDITION DEPENDENT CONSTRAINTS$!
1278 5:$.....$!
1279 5:$
1280 5: FOR BC = 1 TO NUMOPTBC DO !
1281 6: CALL ABOUND ( NITER, BC, CONST, ACTBOUND, NAUS, NACSD, [PGAS], !
1282 6: PCAS, ACTAERO, ACTDYN, ACTFLUT, !
1283 6: NMPC, NSPC, NOMIT, NRSET, NGDR, USET(BC) ); !
1284 6: IF ACTBOUND THEN !
1285 7:$
1286 7:$ REESTABLISH THE BASE USET AND PARTITIONING DATA FOR THE BC $!
1287 7:$ IF GDR CHANGED IT $!
1288 7:$ NOTE, THIS LEAVES AN INCOMPATIBILITY BETWEEN USET(BC) AND $!
1289 7:$ BGPDT(BC) SINCE THE LATTER IS NOT REGENERATED. $!
1290 7:$ THIS INCOMPATIBILITY WILL NOT AFFECT THE SENSITIVITY ANALYSIS$!
1291 7:$ AND WILL BE CORRECTED IN THE SUBSEQUENT ANALYSIS $!
1292 7:$

```

Standard MAPOL Sequence Listing — Continued

```

STAT LEVL
1293 7: IF NGDR <> 0 THEN
1294 8: CALL MKUSET(BC, GSIZEB, [YS(BC)], [TMN(BC)], [PGMN(BC)],
1295 8: [PNSF(BC)], [PFOA(BC)], [PARL(BC)], USET(BC));
1296 8: ENDF;
1297 7:$
1298 7:$ EVALUATE FREQUENCY CONSTRAINT SENSITIVITIES
1299 7:$
1300 7: IF ACTDYN THEN
1301 8: IF NGDR <> 0 THEN
1302 9: CALL ROWPART ([PHIG(BC)], [GTMP], [PGDRG(BC)]);
1303 9: CALL FREQSNS ( NITER, BC, NDV, GLBDES, CONST, LAMBDA,
1304 9: GMKCT, DKVI, GMMCT, DMVI,
1305 9: [GTMP], [AMAT] );
1306 9: ELSE
1307 9: CALL FREQSNS ( NITER, BC, NDV, GLBDES, CONST, LAMBDA,
1308 9: GMKCT, DKVI, GMMCT, DMVI,
1309 9: [PHIG(BC)], [AMAT] );
1310 9: ENDF;
1311 8: ENDF;
1312 7:$
1313 7:$ EVALUATE FLUTTER CONSTRAINT SENSITIVITIES
1314 7:$
1315 7: IF ACTFLUT THEN
1316 8: SUB := 0;
1317 8: LOOP := TRUE;
1318 8: IF NGDR <> 0 CALL ROWPART ([PHIG(BC)], [GTMP], [PGDRG(BC)]);
1319 8: WHILE LOOP DO
1320 9: SUB := SUB + 1;
1321 9: IF NGDR <> 0 THEN
1322 10: CALL FLUTSENS (NITER, BC, SUB, LOOP, GSIZEB, NDV,
1323 10: GLBDES, CONST, GMKCT, DKVI, GMMCT,
1324 10: DMVI, CLAMBDA, LAMBDA,
1325 10: [QHHFL(BC,SUB)],
1326 10: [MHHFL(BC,SUB)], [BHHFL(BC,SUB)],
1327 10: [KHHFL(BC,SUB)], [GTMP], [AMAT]);
1328 10: ELSE
1329 10: CALL FLUTSENS (NITER, BC, SUB, LOOP, GSIZEB, NDV,
1330 10: GLBDES, CONST, GMKCT, DKVI, GMMCT,
1331 10: DMVI, CLAMBDA, LAMBDA,
1332 10: [QHHFL(BC,SUB)],
1333 10: [MHHFL(BC,SUB)], [BHHFL(BC,SUB)],
1334 10: [KHHFL(BC,SUB)], [PHIG(BC)], [AMAT]);
1335 10: ENDF;
1336 9: ENDDO;
1337 8: ENDF;
1338 7:$
1339 7:$ EVALUATE ACTIVE DISPLACEMENT DEPENDENT CONSTRAINTS FROM
1340 7:$ THE STATICS DISCIPLINE
1341 7:$
1342 7: IF NAUS > 0 THEN
1343 8:$
1344 8:$ SENSITIVITIES OF CONSTRAINTS WRT DISPLACEMENTS FOR STATICS:

```


Standard MAPOL Sequence Listing — Continued

STAT	LEVL		
1345	8!\$		\$!
1346	8!	CALL NULLMAT ([DFDU], [DPGV]);	!
1347	8!	IF NACSD > NAUS * NDV THEN	!
1348	9!\$		\$!
1349	9!\$	USE GRADIENT METHOD	\$!
1350	9!\$		\$!
1351	9!	CALL MAKDFU (NITER, BC, GSIZEB, [SMAT], [GLBSIG],	!
1352	9!	CONST, [DFDU]);	!
1353	9!	ELSE	!
1354	9!\$		\$!
1355	9!\$	USE VIRTUAL LOAD METHOD	\$!
1356	9!\$		\$!
1357	9!	CALL MAKDFU (NITER, BC, GSIZEB, [SMAT], [GLBSIG],	!
1358	9!	CONST, [DPGV]);	!
1359	9!	ENDIF;	!
1360	8!\$		\$!
1361	8!\$	SOME RELATIVELY SIMPLE CALCULATIONS THAT PRECEDE THE	\$!
1362	8!\$	LOOP ON THE DESIGN VARIABLES	\$!
1363	8!\$		\$!
1364	8!	IF NGDR <> 0 THEN	!
1365	9!	CALL PARTN ([UG(BC)],,,, [UGA], [PGAS], [PGDRG(BC)]);	!
1366	9!	ELSE	!
1367	9!	CALL COLPART ([UG(BC)], , [UGA], [PGAS]);	!
1368	9!	ENDIF;	!
1369	8!\$		\$!
1370	8!\$	OBTAIN THE SENSITIVITIES OF THE DESIGN	\$!
1371	8!\$	DEPENDENT LOADS	\$!
1372	8!\$		\$!
1373	8!	CALL DDLOAD(NDV, GSIZEB, BC, SMPLOD, DDFLG,[PGAS],[DPVJ]);	!
1374	8!\$		\$!
1375	8!	CALL MAKDVU (NITER, NDV, GLBDES, [UGA], [DKUG],	!
1376	8!	GMKCT, DKVI);	!
1377	8!	CALL NULLMAT ([DUG]);	!
1378	8!	IF NRSET <> 0 THEN	!
1379	9!	IF NGDR <> 0 THEN	!
1380	10!	CALL PARTN ([AG(BC)],,,, [AGA], [PGAS], [PGDRG(BC)]);	!
1381	10!	ELSE	!
1382	10!	CALL COLPART ([AG(BC)], , [AGA], [PGAS]);	!
1383	10!	ENDIF;	!
1384	9!	CALL MAKDVU (NITER, NDV, GLBDES, [AGA], [DMAG],	!
1385	9!	GMMCT, DMVI);	!
1386	9!	[DUG] := [DKUG] + [DMAG];	!
1387	9!	ELSE	!
1388	9!	[DUG] := [DKUG];	!
1389	9!	ENDIF;	!
1390	8!\$		\$!
1391	8!\$	ACCOUNT FOR VIRTUAL LOAD METHOD	\$!
1392	8!\$		\$!
1393	8!	IF NACSD > NAUS * NDV THEN	!
1394	9!\$		\$!
1395	9!\$	USE GRADIENT METHOD	\$!
1396	9!\$		\$!

Standard MAPOL Sequence Listing — Continued

```

STAT LEVL
1397 9! IF DDFLG > 0 THEN !
1398 10! [DPGV] := [DPVJ] + [DUG]; !
1399 10! ELSE !
1400 10! [DPGV] := [DUG]; !
1401 10! ENDIF; !
1402 9! ELSE !
1403 9!$ USE VIRTUAL LOAD METHOD $!
1404 9!$ !
1405 9!$ !
1406 9! IF DDFLG > 0 THEN !
1407 10! [DFDU] := [DPVJ] + [DUG]; !
1408 10! ELSE !
1409 10! [DFDU] := [DUG]; !
1410 10! ENDIF; !
1411 9! ENDIF; !
1412 8!$ !
1413 8!$ REDUCE THE RIGHT HAND SIDES TO THE L SET $!
1414 8!$ !
1415 8! CALL NULLMAT ( [DPNV], [DMUN] ); !
1416 8! IF NMPC <> 0 THEN !
1417 9! CALL GREduce (, [DPGV], [PGMN(BC)], [TMN(BC)],, [DPNV]); !
1418 9! ELSE !
1419 9! [DPNV] := [DPGV]; !
1420 9! ENDIF; !
1421 8!$ !
1422 8! CALL NULLMAT ( [DPFV], [DMUF] ); !
1423 8! IF NSPC <> 0 THEN !
1424 9! CALL NREDUCE (, [DPNV], [PNSF(BC)],, , , [DPFV]); !
1425 9! ELSE !
1426 9! [DPFV] := [DPGV]; !
1427 9! ENDIF; !
1428 8!$ !
1429 8! CALL NULLMAT ( [DPAV], [DMUA] ); !
1430 8! IF NGDR <> 0 THEN !
1431 9! [DPAV] := TRANS( [GSUBO(BC)] ) * [DPFV]; !
1432 9! ELSE !
1433 9! IF NOMIT <> 0 THEN !
1434 10! CALL FREDUCE (, [DPFV], [PFOA(BC)],, !
1435 10! [KOOINV(BC)],, , [GSUBO(BC)],, !
1436 10! [DPAV], [DPOV], USET(BC) ); !
1437 10! ELSE !
1438 10! [DPAV] := [DPFV]; !
1439 10! ENDIF; !
1440 9! ENDIF; !
1441 8!$ !
1442 8! IF NRSET <> 0 THEN !
1443 9! CALL ROWPART ( [DPAV], [DPRV], [DPLV], [PARL(BC)] ); !
1444 9! [DRHS] := TRANS( [D(BC)] ) * [DPLV] + [DPRV]; !
1445 9!$ !
1446 9!$ PROCESS ACTIVE CONSTRAINTS FOR STATICS DISCIPLINE $!
1447 9!$ !
1448 9! CALL INERTIA ( [MRR(BC)], [DRHS], [DURD] ); !

```

Standard MAPOL Sequence Listing — Continued

```

STAT LEVL
1449 9! [DULD] := [D(BC)] * [DURD]; !
1450 9! CALL ROWMERGE ( [DUAD], [DURD], [DULD], [PARL(BC)] ); !
1451 9! [DPLV] := [DPLV] + [IFR(BC)] * [DURD]; !
1452 9! CALL FBS ( [KLLINV(BC)], [DPLV], [DULV] ); !
1453 9! CALL YSMERGE ( [DUAV], , [DULV], [PARL(BC)] ); !
1454 9! ELSE !
1455 9! CALL FBS ( [KLLINV(BC)], [DPAV], [DUAV] ); !
1456 9! ENDIF; !
1457 8!$ RECOVER TO THE F SET $!
1458 8!$ $!
1459 8!$ $!
1460 8! CALL NULLMAT ( [DUFV] ); !
1461 8! IF NGDR <> 0 THEN !
1462 9! [DUFV] := [GSUBO(BC)] * [DUAV]; !
1463 9! ELSE !
1464 9! IF NOMIT <> 0 THEN !
1465 10! IF NRSET <> 0 THEN !
1466 11! [TMP1] := [DPOV] - [IFM(BC)] * [DUAD]; !
1467 11! ELSE !
1468 11! [TMP1] := [DPOV]; !
1469 11! ENDIF; !
1470 10! CALL FBS ( [KOOINV(BC)], [TMP1], [UOO] ); !
1471 10! [UO] := [GSUBO(BC)] * [DUAV] + [UOO]; !
1472 10! CALL ROWMERGE ([DUFV], [UO], [DUAV], [PFOA(BC)]); !
1473 10! ELSE !
1474 10! [DUFV] := [DUAV]; !
1475 10! ENDIF; !
1476 9! ENDIF; !
1477 8!$ $!
1478 8!$ REDUCE THE LEFT HAND SIDE MATRIX $!
1479 8!$ $!
1480 8! IF NMPC <> 0 THEN !
1481 9! CALL GREduce ( , [DFDU], [PGMN(BC)], [TMN(BC)], , [DFDUN] ); !
1482 9! ELSE !
1483 9! [DFDUN] := [DFDU]; !
1484 9! ENDIF; !
1485 8!$ $!
1486 8! IF NSPC <> 0 THEN !
1487 9! CALL ROWPART ( [DFDUN], , [DFDUF], [PNSF(BC)] ); !
1488 9! ELSE !
1489 9! [DFDUF] := [DFDUN]; !
1490 9! ENDIF; !
1491 8!$ $!
1492 8!$ ACCOUNT FOR VIRTUAL LOAD METHOD $!
1493 8!$ $!
1494 8! IF NACSD > NAUS * NDV THEN !
1495 9!$ $!
1496 9!$ USE GRADIENT METHOD $!
1497 9!$ $!
1498 9! CALL MKAMAT ([AMAT], [DFDUF], [DUFV], PCAS, [PGAS] ); !
1499 9! ELSE !
1500 9!$ $!

```

Standard MAPOL Sequence Listing — Continued

```

STAT LEVL
1501 9:$ USE VIRTUAL LOAD METHOD $!
1502 9:$ $!
1503 9: CALL MKAMAT ([AMAT], [DUFV], [DFDUF], PCAS, [PGAS] ); !
1504 9: ENDF; !
1505 8:$ $!
1506 8: ENDF; $ END IF ON ACTIVE APPLIED STATIC LOADS $!
1507 7:$ $!
1508 7:$ EVALUATE ACTIVE CONSTRAINTS FROM $!
1509 7:$ THE STATIC AEROELASTICITY DISCIPLINE $!
1510 7:$ $!
1511 7: IF ACTAERO THEN !
1512 8: LOOP := TRUE; !
1513 8: ACTUAGG := FALSE; !
1514 8: SUB := 0; !
1515 8: CALL NULLMAT ( [DUFV] ); !
1516 8: WHILE LOOP DO !
1517 9: SUB := SUB + 1; !
1518 9: CALL AROSNSDR ( NITER, BC, SUB, LOOP, MINDEX, CONST, !
1519 9: SYM, NGDR, !
1520 9: [PGDRG(BC)], [UAG(BC)], [AAG(BC)], !
1521 9: ACTUAG, [UGA], [AGA], [PGAA], [PGAU], !
1522 9: PCAA, [UAGC(BC,SUB)], [AAGC(BC,SUB)], !
1523 9: ACTAEFF, [AUAGC], [AAAGC], PCAE ); !
1524 9: IF ACTAEFF THEN !
1525 10:$ $!
1526 10:$ PROCESS PSEUDO DISPLACEMENTS FOR EFFECTIVENESS $!
1527 10:$ CONSTRAINTS $!
1528 10:$ $!
1529 10: CALL MAKDVU ( NITER, NDV, GLBDES, [AUAGC], [DKUG], !
1530 10: GMMCT, DKVI ); !
1531 10: IF NRSET <> 0 THEN !
1532 11: CALL MAKDVU ( NITER, NDV, GLBDES, [AAAGC], [DMAG], !
1533 11: GMMCT, DMVI); !
1534 11: [DPGV] := [DKUG] + [DMAG]; !
1535 11: CALL MAKDVU ( NITER, NDV, GLBDES, [AUAGC], [DMUG], !
1536 11: GMMCT, DMVI); !
1537 11: ELSE !
1538 11: [DPGV] := [DKUG]; !
1539 11: ENDF; !
1540 10:$ $!
1541 10:$ REDUCE THE RIGHT HAND SIDES TO THE L SET $!
1542 10:$ $!
1543 10: CALL NULLMAT ( [DPNV], [DMUN] ); !
1544 10: IF NMPC <> 0 THEN !
1545 11: CALL GREduce ( , [DPGV], [PGMN(BC)], [TMN(BC)], !
1546 11: [DPNV]); !
1547 11: IF NRSET <> 0 CALL GREduce ( , [DMUG], !
1548 12: [PGMN(BC)], [TMN(BC)], [DMUN] ); !
1549 11: ELSE !
1550 11: [DPNV] := [DPGV]; !
1551 11: IF NRSET <> 0 [DMUN] := [DMUG]; !
1552 11: ENDF; !

```

Standard MAPOL Sequence Listing — Continued

```

STAT LEVL
1553 10:$
1554 10:
1555 10:
1556 11:
1557 11:
1558 12:
1559 11:
1560 11:
1561 11:
1562 11:
1563 10:$
1564 10:
1565 10:
1566 11:
1567 11:
1568 11:
1569 11:
1570 12:
1571 12:
1572 12:
1573 12:
1574 12:
1575 13:
1576 13:
1577 13:
1578 13:
1579 12:
1580 12:
1581 12:
1582 12:
1583 11:
1584 10:$
1585 10:
1586 11:
1587 11:
1588 11:
1589 11:
1590 11:
1591 11:
1592 11:
1593 11:
1594 11:$
1595 11:$
1596 11:$
1597 11:
1598 11:
1599 11:
1600 11:$
1601 11:
1602 11:
1603 11:$
1604 11:

CALL NULLMAT ( [DPFV], [DMUF] );
IF NSPC <> 0 THEN
  CALL NREDUCE (,[DPNV],[PNSF(BC)],,,,, [DPFV]);
  IF NKSET <> 0
    CALL NREDUCE (,[DMUN],[PNSF(BC)],,,,, [DMUF]);
ELSE
  [DPFV] := [DPGV];
  IF NRSET <> 0 [DMUF] := [DMUN];
ENDIF;
$!

CALL NULLMAT ( [DPAV], [DMUA] );
IF NGDR <> 0 THEN
  [DPAV] := TRANS( [GSUBO(BC)] ) * [DPFV];
  IF NRSET <> 0 [DMUA] := TRANS([GSUBO(BC)]) * [DMUF];
ELSE
  IF NOMIT <> 0 THEN
    CALL FREDUCE ( , [DPFV], [PFOA(BC)], 1,
      [KOOL(BC,SUB)], [KOOU(BC,SUB)],
      [KAO(BC,SUB)], [GASUBO(BC,SUB)], ,
      [DPAV], [DPOV], USET(BC) );
    IF NRSET <> 0
      CALL FREDUCE ( , [DMUF], [PFOA(BC)], 1,
        [KOOL(BC,SUB)], [KOOU(BC,SUB)],
        [KAO(BC,SUB)], [GASUBO(BC,SUB)], ,
        [DMUA], [DMUO], USET(BC) );
  ELSE
    [DPAV] := [DPFV];
    IF NRSET <> 0 [DMUA] := [DMUF];
  ENDIF;
ENDIF;
$!

IF NRSET <> 0 THEN
  CALL ROWPART ([DPAV],[DPRV],[DPLV],[PARL(BC)] );
  CALL ROWPART ([DMUA],[DMUR],[DMUL],[PARL(BC)] );
  CALL GFBS ( [RL11(BC,SUB)], [RU11(BC,SUB)],
    [DPLV], [R11DPL] );
  [DP1] := TRANS([D(BC)]) * [DMUL] + [DMUR] -
    [R21(BC,SUB)] * [R11DPL];
  [DRHS] := TRANS( [D(BC)] ) * [DPLV] + [DPRV] -
    [R31(BC,SUB)] * [R11DPL];
$!

PROCESS ACTIVE CONSTRAINTS FOR SAERO DISCIPLINE
$!

CALL GFBS ( [KL11(BC,SUB)], [KU11(BC,SUB)],
  [DP1], [DK1V] );
[DRHS] := [DRHS] - [K21(BC,SUB)] * [DK1V];
$!

CALL DECOMP ( [LHSA(BC,SUB)], [LHSL], [LHSU] );
CALL GFBS ( [LHSL], [LHSU], [DRHS], [DU2] );
$!

[DU1R] := [DK1V] + [K1112(BC,SUB)] * [DU2];

```

Standard MAPOL Sequence Listing — Continued

```

STAT LEVEL
1605 11! [DU1L] := [R11DPL] + [R1112(BC,SUB)] * [DU1R] +!
1606 11! [R1113(BC,SUB)] * [DU2]; !
1607 11! [EFFSENS] := - [R31(BC,SUB)] * [DU1L] - !
1608 11! [R32(BC,SUB)] * [DU1R]; !
1609 11!$ $!
1610 11! CALL AEROEFFS ( NITER, BC, SUB, SYM, NDV, CONST, !
1611 11! PCAE, [EFFSENS], [AMAT] ); !
1612 11! ELSE !
1613 11!$ $!
1614 11!$ NOTE THAT SAERO W/O SUPPORT IS NOT SUPPORTED $!
1615 11!$ $!
1616 11!$ ENDIF; $!
1617 10!$ ENDIF; $ END IF ON ACTAEFF $!
1618 9!$ $!
1619 9! IF ACTUAG THEN !
1620 10!$ $!
1621 10!$ SENSITIVITIES OF CONSTRAINTS WRT DISPLACEMENTS $!
1622 10!$ FOR SAERO. THE ACTUAGG FLAG WILL BE RETURNED $!
1623 10!$ FALSE IF ONLY TRIM PARAMETER CONSTRAINTS ARE ACTIVE $!
1624 10!$ $!
1625 10! CALL NULLMAT ( [DFDU] ); !
1626 10! CALL MAKDFU ( NITER, BC, GSIZEB, [SMAT], [GLBSIG], !
1627 10! CONST, [DFDU], ACTUAGG, SUB ); !
1628 10!$ $!
1629 10!$ SOME RELATIVELY SIMPLE CALCULATIONS THAT PRECEDE $!
1630 10!$ THE LOOP ON THE DESIGN VARIABLES $!
1631 10!$ $!
1632 10! CALL MAKDVU ( NITER, NDV, GLBDES, [UGA], [DKUG], !
1633 10! GMMCT, DKVI ); !
1634 10! CALL NULLMAT ( [DPGV] ); !
1635 10! IF NRSET <> 0 THEN !
1636 11! CALL MAKDVU ( NITER, NDV, GLBDES, [AGA], [DMAG], !
1637 11! GMMCT, DMVI ); !
1638 11! [DPGV] := [DKUG] + [DMAG]; !
1639 11! CALL MAKDVU ( NITER, NDV, GLBDES, [UGA], [DMUG], !
1640 11! GMMCT, DMVI ); !
1641 11! ELSE !
1642 11! [DPGV] := [DKUG]; !
1643 11!$ ENDIF; $!
1644 10!$ $!
1645 10!$ REDUCE THE RIGHT HAND SIDES TO THE L SET $!
1646 10!$ $!
1647 10! CALL NULLMAT ( [DPNV], [DMUN] ); !
1648 10! IF NMPC <> 0 THEN !
1649 11! CALL GREduce ( , [DPGV], [PGMN(BC)], [TMN(BC)], !
1650 11! [DPNV] ); !
1651 11! IF NRSET <> 0 CALL GREduce ( , [DMUG], !
1652 12! [PGMN(BC)], [TMN(BC)], [DMUN] ); !
1653 11! ELSE !
1654 11! [DPNV] := [DPGV]; !
1655 11! IF NRSET <> 0 [DMUN] := [DMUG]; !
1656 11!$ ENDIF; $!

```

Standard MAPOL Sequence Listing — Continued

```

STAT LEVL
1657 10:$
1658 10: CALL NULLMAT ( [DPFV], [DMUF] ); $!
1659 10: IF NSPC <> 0 THEN !
1660 11: CALL NREDUCE (,[DPNV],[PNSF(BC)],..., [DPFV]); !
1661 11: IF NPSET <> 0 !
1662 12: CALL NREDUCE (,[DMUN],[PNSF(BC)],..., [DMUF]); !
1663 11: ELSE !
1664 11: [DPFV] := [DPGV]; !
1665 11: IF NRSET <> 0 [DMUF] := [DMUN]; !
1666 11: ENDIF; !
1667 10:$ $!
1668 10: CALL NULLMAT ( [DPAV], [DMUA] ); !
1669 10: IF NGDR <> 0 THEN !
1670 11: [DPAV] := TRANS( [GSUBO(BC)] ) * [DPFV]; !
1671 11: IF NPSET <> 0 [DMUA]:=TRANS([GSUBO(BC)])*[DMUF]; !
1672 11: ELSE !
1673 11: IF NOMIT <> 0 THEN !
1674 12: CALL FREDUCE ( , [DPFV], [PFOA(BC)], 1, !
1675 12: [KOOL(BC,SUB)], [KOOU(BC,SUB)], !
1676 12: [KAO(BC,SUB)], [GASUBO(BC,SUB)], , !
1677 12: [DPAV], [DPOV], USET(BC) ); !
1678 12: IF NRSET <> 0 !
1679 13: CALL FREDUCE ( , [DMUF], [PFOA(BC)], 1, !
1680 13: [KOOL(BC,SUB)], [KOOU(BC,SUB)], !
1681 13: [KAO(BC,SUB)], [GASUBO(BC,SUB)], , !
1682 13: [DMUA], [DMUO], USET(BC) ); !
1683 12: ELSE !
1684 12: [DPAV] := [DPFV]; !
1685 12: IF NRSET <> 0 [DMUA] := [DMUF]; !
1686 12: ENDIF; !
1687 11: ENDIF; !
1688 10:$ $!
1689 10: IF NRSET <> 0 THEN !
1690 11: CALL ROWPART ([DPAV],[DPRV],[DPLV],[PARL(BC)] ); !
1691 11: CALL ROWPART ([DMUA],[DMUR],[DMUL],[PARL(BC)] ); !
1692 11: CALL GFBS ( [RL11(BC,SUB)], [RU11(BC,SUB)], !
1693 11: [DPLV], [R11DPL] ); !
1694 11: [DPL] := TRANS([D(BC)]) * [DMUL] + [DMUR] - !
1695 11: [R21(BC,SUB)] * [R11DPL]; !
1696 11: [DRHS] := TRANS( [D(BC)] ) * [DPLV] + [DPRV] - !
1697 11: [R31(BC,SUB)] * [R11DPL]; !
1698 11:$ $!
1699 11:$ PROCESS ACTIVE CONSTRAINTS FOR SAERO DISCIPLINE $!
1700 11:$ $!
1701 11: CALL GFBS ( [KL11(BC,SUB)], [KU11(BC,SUB)], !
1702 11: [EP1], [DK1V] ); !
1703 11: [DRHS] := [DRHS] - [K21(BC,SUB)] * [DK1V]; !
1704 11:$ $!
1705 11: CALL AEROSENS ( NITER, BC, MINDEX, SUB, CONST, !
1706 11: SYM, NDV, !
1707 11: BGPDT(BC), STABCF, [PGAA], !
1708 11: [LHSA(BC,SUB)], [RHSA(BC,SUB)], !

```

Standard MAPOL Sequence Listing — Continued

```

STAT LEVL
1709 11! [DRHS], [AAR], [DDELDV], [AMAT] );!
1710 11!$ $!
1711 11! [DURV] := [K1112(BC,SUB)] * [AAR] + !
1712 11! [PAR(BC,SUB)] * [DDELDV] + [DK1V]; !
1713 11! [DULV] := [R1112(BC,SUB)] * [DURV] + !
1714 11! [R1113(BC,SUB)] * [AAR] - !
1715 11! [R11PAL(BC,SUB)] * [DDELDV] + [R11DPL]; !
1716 11! CALL ROWMERGE ([DUAV],[DURV],[DULV],[PAR(BC)]); !
1717 11! ELSE !
1718 11!$ $!
1719 11!$ NOTE THAT SAERO W/O SUPPORT IS NOT SUPPORTED $!
1720 11!$ $!
1721 11! ENDF; !
1722 10!$ $!
1723 10!$ RECOVER SENSITIVITIES TO THE F SET $!
1724 10!$ $!
1725 10! CALL NULLMAT ([UAFTMP] ); !
1726 10! IF NGDR <> 0 THEN !
1727 11! [UAFTMP] := [GASUBO(BC,SUB)] * [DUAV]; !
1728 11! ELSE !
1729 11! IF NOMIT <> 0 THEN !
1730 12! IF NRSET <> 0 THEN !
1731 13! [TMP1] := [DPOV]+[POARO(BC,SUB)]*[DDELDV]; !
1732 13! ELSE !
1733 13! [TMP1] := [DPOV]; !
1734 13! ENDF; !
1735 12! CALL GFBS ([KOOL(BC,SUB)], [KOOO(BC,SUB)], !
1736 12! [TMP1], [UOO]); !
1737 12! [UO] := [GASUBO(BC,SUB)] * [DUAV] + [UOO]; !
1738 12! CALL ROWMERGE ([UAFTMP], [UO], [DUAV], !
1739 12! [PFOA(BC)] ); !
1740 12! ELSE !
1741 12! [UAFTMP] := [DUAV]; !
1742 12! ENDF; !
1743 11! ENDF; !
1744 10! CALL AROSNSMR ( BC, SUB, NDV, [PGAA], [PGAU], [DUFV], !
1745 10! [UAFTMP] ); !
1746 10!$ $!
1747 10! ENDF; $ END IF ON ACTUAG $!
1748 9! ENDDO; $ END DO ON SUBSCRIPT LOOP $!
1749 8!$ $!
1750 8! IF ACTUAGG THEN !
1751 9!$ $!
1752 9!$ REDUCE THE LEFT HAND SIDE MATRIX $!
1753 9!$ $!
1754 9! CALL NULLMAT ([DFDUN] ); !
1755 9! IF NMPC <> 0 THEN !
1756 10! CALL GREduce ( , [DFDU], [PGMN(BC)], [TMN(BC)], !
1757 10! [DFDUN]); !
1758 10! ELSE !
1759 10! [DFDUN] := [DFDU]; !
1760 10! ENDF; !

```


Standard MAPOL Sequence Listing — Continued

```

STAT LEVL
1761 9!$
1762 9! CALL NULLMAT ( [DFDUF] );
1763 9! IF NSPC <> 0 THEN
1764 10! CALL ROWPART ( [DFDUN], , [DFDUF], [PNSF(BC)] );
1765 10! ELSE
1766 10! [DFDUF] := [DFDUN];
1767 10! ENDIF;
1768 9!$
1769 9!$ TAKE MERGED SENSITIVITIES OF DISPLACEMENTS AND
1770 9!$ COMPUTE THE AMAT MATRIX TERMS FOR THE SAERO
1771 9!$ CONSTRAINTS
1772 9!$
1773 9! CALL MKAMAT ([AMAT], [DFDUF], [DUFV], PCAA, [PGAU] );
1774 9!$
1775 9! ENDIF; $ END IF ON ANY ACTIVE DISPLACEMENTS
1776 8! ENDIF; $ END IF ON ACTIVE AEROELASTIC CONSTRAINTS
1777 7! ENDIF; $ END IF ON ACTIVE BOUNDARY CONDITION
1778 6! ENDDO; $ END DO ON ACTIVE BOUNDARY CONDITIONS
1779 5!$
1780 5! CALL OFPGRAD ( NITER, NUMOPTBC, [AMAT], GLBDES, CONST, GRADIENT );
1781 5!$
1782 5! IF NITER >= OCS AND NITER <= OCE THEN
1783 6! PRINT("LOG=(' VANGO MODULE')");
1784 6! CALL VANGO ( NITER, NDV, APPCNVRG, MOVLIM, CNVRGLIM,
1785 6! CTL, CTLMIN, NUMOPTBC, GLBDES, CONST, [AMAT],
1786 6! DESHIST );
1787 6! ELSE
1788 6! IF NITER >= MPS AND NITER <= MPE THEN
1789 7! PRINT("LOG=(' DESIGN MODULE')");
1790 7! CALL DESIGN( NITER, NDV, APPCNVRG, MOVLIM, CNVRGLIM,
1791 7! CTL, CTLMIN, NUMOPTBC, GLBDES, CONST, [AMAT],
1792 7! DESHIST );
1793 7! ENDIF;
1794 6! ENDIF;
1795 5!$
1796 5! ENDIF; $ END IF ON FSD METHOD
1797 4! ENDIF; $ END IF TEST AFTER ACTCON
1798 3! ENDDO; $ END WHILE LOOP FOR GLOBAL CONVERGENCE
1799 2!ENDIF; $ END IF ON OPTIMIZATION
1800 1!$
1801 1!$*****$
1802 1!$ BEGIN FINAL ANALYSIS LOOP
1803 1!$*****$
1804 1!$
1805 1!IF NBNDCOND > NUMOPTBC THEN
1806 2!$
1807 2!$ ASSEMELE THE GLOBAL MATRICES
1808 2!$
1809 2! PRINT("LOG=('*****')");
1810 2!$
1811 2!$ ASSEMBLE THE GLOBAL MATRICES
1812 2!$ BEGIN BOUNDARY CONDITION LOOP

```

Standard MAPOL Sequence Listing — Continued

```

STAT LEVL
1813 2:$ PRINT('LOG= ('BEGIN FINAL ANALYSIS')'); $!
1814 2: CALL ANALINIT; !
1815 2: CALL EMA2 ( , NDV, GSIZEB, GLBDES, GMMCT, DKVI, [K1GG], !
1816 2: GMMCT, DMVI, [M1GG] ); !
1817 2: FOR BC = NUMOPTBC + 1 TO NBNDCOND DO !
1818 2: PRINT('LOG= (' BOUNDARY CONDITION ',I3)',BC); !
1819 3:$ ESTABLISH THE BASE USET AND PARTITIONING DATA FOR THE BC $!
1820 3:$ CALL MKUSET( BC, GSIZEB, [YS(BC)], [TMN(BC)], [PGMN(BC)], [PNSF(BC)], $!
1821 3:$ [PFOA(BC)], [PARL(BC)], USET(BC) ); !
1822 3:$ MAKE B.C.-DEPENDENT BGPDT FROM BASE, ADDING THE EXTRA POINTS FOR $!
1823 3:$ THIS B.C. $!
1824 3:$ CALL BCBGPD( BC , GSIZEB , BGPDT(BC) , ESIZE(BC) ); !
1825 3:$ GSIZE := GSIZEB; !
1826 3:$ PSIZE(BC) := ESIZE(BC) + GSIZE; !
1827 3:$ PROCESS MATRICES, TRANSFER FUNCTIONS, AND INITIAL CONDITIONS FOR $!
1828 3:$ THIS B.C. $!
1829 3:$ CALL BCBULK( BC , PSIZE(BC) , BGPDT(BC) , USET(BC) ); !
1830 3:$ CALL BOUND ( BC, GSIZE, ESIZE(BC), USET(BC), BLOAD, BMASS, DMODES, !
1831 3:$ BMODES, BSAERO, BFLUTR, BDYN, BDRSP, BDTR, BNTR, BDFR, !
1832 3:$ BMFR, BGUST, BBLAST, NMPC, NSPC, NOMIT, NRSET, NGDR ); !
1833 3:$ DETERMINE IF ANY M2GG/K2GG INPUT DATA ARE TO BE ADDED $!
1834 3:$ CALL NULLMAT ( [KGG], [MGG] ); !
1835 3:$ CALL MK2GG ( BC, GSIZEB, [M2GG], M2GGFLAG, [K2GG], K2GGFLAG ); !
1836 3:$ IF M2GGFLAG THEN !
1837 4:$ [MGG] := [M1GG] + [M2GG]; !
1838 4:$ ELSE !
1839 4:$ [MGG] := [M1GG]; !
1840 4:$ ENDIF; !
1841 3:$ IF K2GGFLAG THEN !
1842 4:$ [KGG] := [K1GG] + [K2GG]; !
1843 4:$ ELSE !
1844 4:$ [KGG] := [K1GG]; !
1845 4:$ ENDIF; !
1846 3:$ CALL THE GRID POINT WEIGHT GENERATOR FOR THIS BOUNDARY CONDITON $!
1847 3:$ CALL GPWG ( , BC, GPWGGRID, [MGG], OGPWG ); !
1848 3:$ IF BLOAD <> 0 CALL GTLOAD ( , BC, GSIZE, BGPDT(BC), GLBDES, !
1849 4:$ SMPLOD, [DPHVI], [DPGRVI], [PG], OGRIDLOD); !
1850 3:$ PARTITION-REDUCTION OF GLOBAL MATRICES $!
1851 3:$

```

Standard MAPOL Sequence Listing — Continued

```

STAT LEVL
1865 3!$
1866 3! IF NBNDCOND > 1 CALL NULLMAT ( [KNN], [PN], [MNN], [GTKN], [GSTKN],
1867 4! [UGTKN] );
1868 3! IF NMPC <> 0 THEN
1869 4!$
1870 4!$ PERFORM MPC REDUCTION
1871 4!$
1872 4! PRINT("LOG=(' MPC REDUCTION')");
1873 4! CALL GREduce ( [KGG], [PG], [PGMN(BC)], [TMN(BC)], [KNN], [PN] );
1874 4! IF BMAS <> 0 CALL GREduce ([MGG], [PGMN(BC)], [TMN(BC)], [MNN]);
1875 4! IF BSAERO <> 0 THEN
1876 5! CALL GREduce (, [GTKG], [PGMN(BC)], [TMN(BC)],, [GTKN]);
1877 5! CALL GREduce (, [GSTKG], [PGMN(BC)], [TMN(BC)],, [GSTKN]);
1878 5! ENDIF;
1879 4! IF BFLUTR <> 0 OR BGUST <> 0 OR BBLAST <> 0
1880 5! CALL GREduce (, [UGTKG], [PGMN(BC)], [TMN(BC)],, [UGTKN] );
1881 4! ELSE
1882 4!$
1883 4!$ NO MPC REDUCTION
1884 4!$
1885 4! [KNN] := [KGG];
1886 4! IF BLOAD <> 0 [PN] := [PG];
1887 4! IF BMAS <> 0 [MNN] := [MGG];
1888 4! IF BSAERO <> 0 THEN
1889 5! [GTKN] := [GTKG];
1890 5! [GSTKN] := [GSTKG];
1891 5! ENDIF;
1892 4! IF BFLUTR <> 0 OR BGUST <> 0 OR BBLAST <> 0 [UGTKN] := [UGTKG];
1893 4! ENDIF;
1894 3! IF NBNDCOND > 1 CALL NULLMAT ( [KFF], [PF], [MFF], [GTKF], [GSTKF],
1895 4! [UGTKF] );
1896 3! IF NSPC <> 0 THEN
1897 4!$
1898 4!$ PERFORM SPC REDUCTION
1899 4!$
1900 4! PRINT("LOG=(' SPC REDUCTION')");
1901 4! CALL NREDUCE ( [KNN], [PN], [PNSF(BC)], [YS(BC)], [KFF], [KFS],
1902 4! [KSS], [PF], [PS] );
1903 4! IF BMAS <> 0 CALL NREDUCE ( [MNN],, [PNSF(BC)],, [MFF]);
1904 4! IF BSAERO <> 0 THEN
1905 5! CALL NREDUCE (, [GTKN], [PNSF(BC)],, [GTKF] );
1906 5! CALL NREDUCE (, [GSTKN], [PNSF(BC)],, [GSTKF] );
1907 5! ENDIF;
1908 4! IF BFLUTR <> 0 OR BGUST <> 0 OR BBLAST <> 0
1909 5! CALL NREDUCE (, [UGTKN], [PNSF(BC)],, [UGTKF]);
1910 4! ELSE
1911 4!$
1912 4!$ NO SPC REDUCTION
1913 4!$
1914 4! [KFF] := [KNN];
1915 4! IF BLOAD <> 0 [PF] := [PN];
1916 4! IF BMAS <> 0 [MFF] := [MNN];

```

Standard MAPOL Sequence Listing — Continued

```

STAT LEVL
1917 4!      IF BSAERO <> 0 THEN                                !
1918 5!          [GTKF] := [GTKN];                                !
1919 5!          [GSTKF] := [GSTKN];                                !
1920 5!      ENDIF;                                              !
1921 4!          IF BFLUTR <> 0 OR BGUST <> 0 OR BBLAST <> 0 [UGTKF] := [UGTKN];    !
1922 4!      ENDIF;                                              !
1923 3!$                                           $!
1924 3!      IF NBNDCOND > 1 CALL NULLMAT ([KAA], [PA], [MAA], [KAAA], [PAA], [UGTKA]); !
1925 3!$                                           $!
1926 3!      IF NGDR <> 0 THEN                                !
1927 4!$                                           $!
1928 4!$      PERFORM THE GENERAL DYNAMIC REDUCTION WHICH IS DISCIPLINE    $!
1929 4!$      INDEPENDENT. THE RESULTING [GSUBO] MATRIX WILL BE USED BY    $!
1930 4!$      ALL DISCIPLINES                                           $!
1931 4!$                                           $!
1932 4!      PRINT('LOG=('          DYNAMIC REDUCTION')');          !
1933 4!$                                           $!
1934 4!$      OBTAIN THE OMITTED DOF PARTITION OF KFF AND MFF          $!
1935 4!$                                           $!
1936 4!      CALL PARTN ( [KFF], [KOO], , [KOA], , [PFOA(BC)] );      !
1937 4!      CALL PARTN ( [MFF], [MOO], , , [PFOA(BC)] );          !
1938 4!      ASIZE := GSIZE - NMPC - NSPC - NOMIT;                  !
1939 4!      LSIZE := ASIZE - NRSET;                                  !
1940 4!      CALL GDR1 ( [KOO], [MOO], [KSOO], [GGO], LKSET, LJSET, NEIV, !
1941 4!          FMAX, BC, BGPDT(BC), USET(BC), NOMIT, LSIZE );      !
1942 4!$                                           $!
1943 4!$      LKSET          MEANING                                $!
1944 4!$      <> 0          APPROX. MODE SHAPES SELECTED            $!
1945 4!$      = 0          NO APPROX. MODE SHAPES IN GDR            $!
1946 4!$                                           $!
1947 4!      IF LKSET <> 0 THEN                                !
1948 5!          CALL SDCOMP ( [KSOO], [LSOO], USET(BC), SINGOSET );    !
1949 5!          CALL GDR2 ( [LSOO], [MOO], [PHIOK], LKSET, LJSET,      !
1950 5!              NEIV, FMAX, BC );                                !
1951 5!      ENDIF;                                              !
1952 4!      CALL GDR3 ( [KOO], [KOA], [MGG], [PHIOK], [TMN(BC)], [GGO], !
1953 4!          [PGMN(BC)], [PNSF(BC)], [PFOA(BC)], [GSUBO(BC)],      !
1954 4!          BGPDT(BC), USET(BC),                                !
1955 4!          LKSET, LJSET, ASIZE, GNORM, BC );                    !
1956 4!      CALL GDR4 ( BC, GSIZE, PSIZE(BC), LKSET, LJSET, NUMOPTBC, NBNDCOND, !
1957 4!          [PGMN(BC)], [TMN(BC)], [PNSF(BC)], [PFOA(BC)],      !
1958 4!          [PARL(BC)], [PGDRG(BC)], [PAJK], [PFJK], BGPDT(BC),    !
1959 4!          USET(BC) );                                           !
1960 4!      ENDIF;                                              !
1961 3!$                                           $!
1962 3!      IF BLOAD <> 0 OR BMODES <> 0 OR BFLUTR <> 0 OR BDYN <> 0 THEN    !
1963 4!$                                           $!
1964 4!$      REDUCE THE MATRICES WITHOUT AEROELASTIC CORRECTIONS        $!
1965 4!$                                           $!
1966 4!      IF NGDR <> 0 THEN                                !
1967 5!$                                           $!
1968 5!$      PERFORM THE GENERAL DYNAMIC REDUCTION                    $!

```

Standard MAPOL Sequence Listing — Continued

```

STAT LEVL
1969 5!$ PRINT("LOG=(' SYMMETRIC DYNAMIC REDUCTION')"); $!
1970 5! !
1971 5!$ $!
1972 5! [MAA] := TRANS ( [GSUBO(BC)] ) * [ [MFF] * [GSUBO(BC)] ]; !
1973 5! [KAA] := TRANS ( [GSUBO(BC)] ) * [ [KFF] * [GSUBO(BC)] ]; !
1974 5! IF BLOAD <> 0 [PA] := TRANS ( [GSUBO(PC)] ) * [PF]; !
1975 5! IF BFLUTR <> 0 OR BGUST <> 0 OR BBLAST <> 0 THEN !
1976 6! [TMP1] := TRANS ( [UGTKF] ) * [GSUBO(BC)]; !
1977 6! CALL TRNSPOSE ( [TMP1], [UGTKA] ); !
1978 6! ENDIF; !
1979 5! ELSE !
1980 5! IF NOMIT <> 0 THEN !
1981 6!$ PERFORM THE STATIC REDUCTION $!
1982 6!$ $!
1983 6!$ $!
1984 6! PRINT("LOG=(' STATIC CONDENSATION')"); !
1985 6!$ $!
1986 6! CALL FREDUCE ( [KFF], [PF], [PFOA(BC)], , [KOOINV(BC)], , !
1987 6! [GSUBO(BC)], [KAA], [PA], [PO], USET(BC) ); !
1988 6!$ $!
1989 6! IF BMASS <> 0 THEN !
1990 7!$ PERFORM GUYAN REDUCTION OF THE MASS MATRIX $!
1991 7!$ $!
1992 7!$ CALL PARTN ( [MFF], [MOO], , [MOA], [MAABAR], [PFOA(BC)] ); !
1993 7! [MAA] := [MAABAR] + TRANS([MOA]) * [GSUBO(BC)] + !
1994 7! TRANS([GSUBO(BC)]) * [MOA] + !
1995 7! TRANS([GSUBO(BC)]) * [ [MOO] * [GSUBO(BC)] ]; !
1996 7! IF NRSET <> 0 [IFM(BC)] := [MOO] * [GSUBO(BC)] + [MOA]; !
1997 7! ENCIF; !
1998 7! IF BFLUTR <> 0 OR BGUST <> 0 OR BBLAST <> 0 THEN !
1999 6! CALL ROWPART ( [UGTKF], [UGTKO], [UGTKAB], [PFOA(BC)] ); !
2000 7! [TMP1] := TRANS( [UGTKO] ) * [GSUBO(BC)]; !
2001 7! CALL TRNSPOSE ( [TMP1], [TMP2] ); !
2002 7! [UGTKA] := [UGTKAB] + [TMP2]; !
2003 7! ENCIF; !
2004 7! ELSE !
2005 6!$ NO F-SET REDUCTION $!
2006 6!$ $!
2007 6!$ $!
2008 6!$ [KAA] := [KFF]; !
2009 6! IF BLOAD <> 0 [PA] := [PF]; !
2010 6! IF BFLUTR <> 0 OR BGUST <> 0 OR BBLAST <> 0 [UGTKA]:=[UGTKF]; !
2011 6! IF BMASS <> 0 [MAA] := [MFF]; !
2012 6! ENCIF; !
2013 6! ENCIF; !
2014 5!$ $!
2015 4!$ IF NRSET <> 0 THEN !
2016 4! !
2017 5!$ PERFORM THE SUPPORT SET REDUCTION $!
2018 5!$ $!
2019 5!$ $!
2020 5! PRINT("LOG=(' SUPPORT REDUCTION')"); !

```

Standard MAPOL Sequence Listing — Continued

```

STAT LEVL
2021 5: CALL PARTN ( [KAA], [KRR], [KLR], , [KLL], [PARL(BC)] ); !
2022 5: CALL SDCOMP ( [KLL], [KLLINV(BC)], USET(BC), SINGLSET ); !
2023 5: CALL FBS ( [KLLINV(BC)], [KLR], [D(BC)], -1 ); !
2024 5: CALL RBCHECK ( BC, USET(BC), BGPDT(BC), [D(BC)], [KLL], !
2025 5: [KRR], [KLR] ); !
2026 5:$ !
2027 5:$ CALCULATE THE REDUCED MASS MATRIX $!
2028 5:$ !
2029 5: CALL PARTN ([MAA], [MRRBAR], [MLR], , [MLL], [PARL(BC)]); !
2030 5: [IFR(BC)] := [MLL] * [D(BC)] + [MLR]; !
2031 5: [MRR(BC)] := [MRRBAR] + TRANS ( [MLR] ) * [D(BC)] + !
2032 5: TRANS ( [D(BC)] ) * [IFR(BC)]; !
2033 5: [R22] := TRANS ( [D(BC)] ) * [MLR] + [MRRBAR]; !
2034 5:$ $!
2035 5: IF BLOAD <> 0 THEN !
2036 6:$ !
2037 6:$ PROCESS STATICS WITH INERTIA RELIEF $!
2038 6:$ !
2039 6: PRINT("LOG=(' >>>DISCIPLINE: STATICS(INERTIA RELIEF)')"); !
2040 6: CALL ROWPART ( [PA], [PR], [PLBAR], [PARL(BC)] ); !
2041 6: [LHS(BC)] := [MRR(BC)]; !
2042 6: [RHS(BC)] := TRANS([D(BC)]) * [PLBAR] + [PR]; !
2043 6: CALL INERTIA ( [LHS(BC)], [RHS(BC)], [AR] ); !
2044 6: [AL] := [D(BC)] * [AR]; !
2045 6: CALL ROWMERGE ( [AA], [AR], [AL], [PARL(BC)] ); !
2046 6: [RHS(BC)] := [PLBAR] - [IFR(BC)] * [AR]; !
2047 6: CALL FBS ( [KLLINV(BC)], [RHS(BC)], [UL] ); !
2048 6: CALL YSMERGE ( [UA], , [UL], [PARL(BC)] ); !
2049 6: ENDIF; !
2050 5: IF BMODES <> 0 THEN !
2051 6: PRINT("LOG=(' >>>DISCIPLINE: NORMAL MODES')"); !
2052 6: CALL REIG ( , BC, USET(BC), [KAA], [MAA], [MRR(BC)], !
2053 6: [D(BC)], LAMBDA, [PHIA], [MII], HSIZE(BC) ); !
2054 6: CALL OFFPMROOT ( , BC, NUMOPTBC, LAMBDA ); !
2055 6: ENDIF; !
2056 5: ELSE !
2057 5:$ !
2058 5:$ NO SUPPORT SET REDUCTION $!
2059 5:$ !
2060 5: IF BLOAD <> 0 THEN !
2061 6: PRINT("LOG=(' >>>DISCIPLINE: STATICS')"); !
2062 6: CALL SDCOMP ( [KAA], [KLLINV(BC)], USET(BC), SINGASET ); !
2063 6: CALL FBS ( [KLLINV(BC)], [PA], [UA] ); !
2064 6: ENDIF; !
2065 5: IF BMODES <> 0 THEN !
2066 6: PRINT("LOG=(' >>>DISCIPLINE: NORMAL MODES')"); !
2067 6: CALL REIG ( , BC, USET(BC), [KAA], [MAA], , LAMBDA, !
2068 6: [PHIA], [MII], HSIZE(BC) ); !
2069 6: CALL OFFPMROOT ( , BC, NUMOPTBC, LAMBDA ); !
2070 6: ENDIF; !
2071 5: ENDIF; !
2072 4: ENDIF; !

```

Standard MAPOL Sequence Listing — Continued

```

STAT LEVL
2073 3:$
2074 3! IF BSAERO <> 0 THEN $!
2075 4:$ $!
2076 4:$ PERFORM STATIC AEROELASTIC ANALYSES $!
2077 4:$ $!
2078 4! PRINT("LOG=(' SAERO INITIALIZATION')"); !
2079 4! CALL TRNSPOSE ( [GSTKF], [GSKF] ); !
2080 4! LOOP := TRUE; !
2081 4! SUB := 0; !
2082 4! WHILE LOOP DO !
2083 5! SUB := SUB + 1; !
2084 5! CALL SAERODRV (BC, SUB, LOOP, MINDEX, SYM, MACH, QDP, 1 ); !
2085 5:$ $!
2086 5:$ ADJUST THE KFF MATRIX AND DETERMINE THE RIGID AIR LOADS $!
2087 5:$ $!
2088 5! IF SYM = 1 [AICS] := [GTKF]*[TRANS([AICMAT(MINDEX)])]*[GSKF]]; !
2089 5! IF SYM = -1 [AICS] := [GTKF]*[TRANS([AAICMAT(MINDEX)])]*[GSKF]]; !
2090 5! [PAF] := (QDP) [ [GTKF] * [AIRFRM(MINDEX)] ]; !
2091 5! [KAFF] := [KFF] - (QDP) [AICS]; !
2092 5:$ $!
2093 5:$ REDUCE THE MATRICES WITH AEROELASTIC CORRECTIONS $!
2094 5:$ SAVE THE SUBCASE/BC DEPENDENT DATA FOR SENSITIVITY ANALYSIS $!
2095 5:$ $!
2096 5! IF NGDR <> 0 THEN !
2097 6:$ $!
2098 6:$ PERFORM THE GENERAL DYNAMIC REDUCTION $!
2099 6:$ $!
2100 6! PRINT("LOG=(' SAERO DYNAMIC REDUCTION')"); !
2101 6! [MAAA] := TRANS ( [GSUBO(BC)] ) * [ [MFF] * [GSUBO(BC)] ]; !
2102 6! [KAAA] := TRANS ( [GSUBO(BC)] ) * [ [KAFF] * [GSUBO(BC)] ]; !
2103 6! [PAA] := TRANS ( [GSUBO(BC)] ) * [PAF]; !
2104 6! ELSE !
2105 6! IF NOMIT <> 0 THEN !
2106 7:$ $!
2107 7:$ PERFORM THE STATIC REDUCTION $!
2108 7:$ $!
2109 7! PRINT("LOG=(' SAERO STATIC CONDENSATION')"); !
2110 7:$ $!
2111 7! IF NRSET <> 0 AND SUB = 1 AND BLOAD = 0 AND BMODES = 0 AND !
2112 8! BFLUTR = 0 AND BDMN = 0 THEN !
2113 8:$ $!
2114 8:$ FORM [KAA] ON SO 'D' CAN BE FORMED $!
2115 8:$ $!
2116 8! CALL FREDUCE ([KFF], , [PFOA(BC)], , [KOOINV(BC)], , , !
2117 8! [GSUBO(BC)], [KAA], , , USET(BC) ); !
2118 8! ENDDIF; !
2119 7:$ $!
2120 7! CALL FREDUCE ( [KAFF], [PAF], [PFOA(BC)], BSAERO, !
2121 7! [KOO(BC,SUB)], [KOO(BC,SUB)], !
2122 7! [KAO(BC,SUB)], [GASUBO(BC,SUB)], [KAAA], !
2123 7! [PAA], [POARO(BC,SUB)], USET(BC)); !
2124 7:$ $!

```

Standard MAPOL Sequence Listing — Continued

```

STAT  LEVL
2125  7!      IF BMASS <> 0 THEN
2126  8!$
2127  8!$      PERFORM GUYAN REDUCTION OF THE MASS MATRIX
2128  8!$
2129  8!      CALL PARTN ( [MFF], [MOO], , [MCA], [MAABAR],
2130  8!          [PFOA(BC)] );
2131  8!      [MAAA] := [MAABAR] + TRANS([MOA]) * [GASUBO(BC,SUB)] +
2132  8!          TRANS([GASUBO(BC,SUB)]) * [MOA] +
2133  8!          TRANS([GASUBO(BC,SUB)]) * ([MOO] *
2134  8!          [GASUBO(BC,SUB)]);
2135  8!      IF NRSET <> 0
2136  9!          [IFLA(BC,SUB)] := [MOO]*[GASUBO(BC,SUB)]+[MOA];
2137  8!      ENDIF;
2138  7!      ELSE
2139  7!$
2140  7!$      NO F-SET REDUCTION
2141  7!$
2142  7!      IF NRSET <> 0 AND SUB = 1 AND BLOAD = 0 AND
2143  8!          BMODES = 0 AND BFLUTR = 0 AND BDYN = 0 THEN
2144  8!$
2145  8!$      FORM [KAA] ON FIRST PASS SO [D] CAN BE FORMED
2146  8!$
2147  8!      [KAA] := [KFF];
2148  8!      ENDIF;
2149  7!      [KAAA] := [KAFF];
2150  7!      [MAAA] := [MFF];
2151  7!      [PAA] := [PAF];
2152  7!      ENDIF;
2153  6!      ENDIF;
2154  5!$
2155  5!      IF NRSET <> 0 THEN
2156  6!$
2157  6!$      PERFORM THE SUPPORT SET REDUCTION
2158  6!$
2159  6!      PRINT("LOG-('      SAERO SUPPORT REDUCTION')");
2160  6!$
2161  6!      IF SUB = 1 AND BLOAD = 0 AND BMODES = 0 AND BFLUTR = 0
2162  7!          AND BDYN = 0 THEN
2163  7!$
2164  7!$      [D] WAS NOT COMPUTED FOR NON-SAERO DISCIPLINES SO
2165  7!$      NEED TO COMPUTE IT NOW
2166  7!$
2167  7!      CALL PARTN ( [KAA], [KRR], [KLR], , [KLL], [PARL(BC)] );
2168  7!      CALL SDCOMP ( [KLL], [KLLINV(BC)], USET(BC), SINGLSET );
2169  7!      CALL FBS ( [KLLINV(BC)], [KLR], [D(BC)], -1 );
2170  7!      CALL RBHECK ( BC, USET(BC), BGPOT(BC), [D(BC)], [KLL],
2171  7!          [KRR], [KIR] );
2172  7!      ENDIF;
2173  6!$
2174  6!$      CALCULATE THE REDUCED MASS MATRIX
2175  6!$
2176  6!      CALL PARTN ([MAAA], [MRRBAR], [MLR], , [MLL], [PARL(BC)]);

```


Standard MAPOL Sequence Listing — Continued

```

STAT LEVL
2177 6! [R13(BC,SUB)] := [MLL] * [D(BC)] + [MLR]; !
2178 6! [R33] := [MRRBAR] + TRANS ( [MLR] ) * [D(BC)] + !
2179 6! TRANS ( [D(BC)] ) * [R13(BC,SUB)]; !
2180 6! [R22] := TRANS ( [D(BC)] ) * [MLR] + [MRRBAR]; !
2181 6! CALL TRNSPOSE ( [R13(BC,SUB)], [R21(BC,SUB)] ); !
2182 6!$ $!
2183 6!$ PROCESS STEADY AEROELASTIC DISCIPLINE $!
2184 6!$ $!
2185 6! PRINT("LOG=(' >>>DISCIPLINE: STEADY AERO')"); !
2186 6! CALL PARTN ( [KAAA], [KARR], [R12(BC,SUB)], [KARL], [R11], !
2187 6! [PARL(BC)] ); !
2188 6! [R32(BC,SUB)] := TRANS([D(BC)]) * [R12(BC,SUB)] + [KARR]; !
2189 6! [R31(BC,SUB)] := TRANS([D(BC)]) * [R11] + [KARL]; !
2190 6!$ $!
2191 6! CALL DECOMP ( [R11], [RL11(BC,SUB)], [RU11(BC,SUB)] ); !
2192 6!$ $!
2193 6! CALL ROWPART ( [PAA], [PARBAR], [PAL], [PARL(BC)] ); !
2194 6! CALL GFBS ( [RL11(BC,SUB)], [RU11(BC,SUB)], [PAL], !
2195 6! [R11PAL(BC,SUB)], -1); !
2196 6! [PRIGID] := [PARBAR] + TRANS([D(BC)]) * [PAL]; !
2197 6! [P1] := [R21(BC,SUB)] * [R11PAL(BC,SUB)]; !
2198 6! [P2] := [PRIGID] + [R31(BC,SUB)] * [R11PAL(BC,SUB)]; !
2199 6!$ $!
2200 6! CALL GFBS ( [RL11(BC,SUB)], [RU11(BC,SUB)], [R12(BC,SUB)], !
2201 6! [R1112(BC,SUB)], -1); !
2202 6! CALL GFBS ( [RL11(BC,SUB)], [RU11(BC,SUB)], [R13(BC,SUB)], !
2203 6! [R1113(BC,SUB)], -1); !
2204 6! [K11] := [R22] + [R21(BC,SUB)] * [R1112(BC,SUB)]; !
2205 6! [K12(BC,SUB)] := [R21(BC,SUB)] * [R1113(BC,SUB)]; !
2206 6! [K21(BC,SUB)] := [R32(BC,SUB)] + [R31(BC,SUB)] * [R1112(BC,SUB)]; !
2207 6! [K22] := [R33] + [R31(BC,SUB)] * [R1113(BC,SUB)]; !
2208 6!$ $!
2209 6! CALL DECOMP ( [K11], [KL11(BC,SUB)], [KU11(BC,SUB)] ); !
2210 6! CALL GFBS ( [KL11(BC,SUB)], [KU11(BC,SUB)], [P1], !
2211 6! [PAR(BC,SUB)] ); !
2212 6! CALL GFBS ( [KL11(BC,SUB)], [KU11(BC,SUB)], [K12(BC,SUB)], !
2213 6! [K1112(BC,SUB)], -1 ); !
2214 6! [LHSA(BC,SUB)] := [K22] + [K21(BC,SUB)] * [K1112(BC,SUB)]; !
2215 6! [RHSA(BC,SUB)] := [P2] - [K21(BC,SUB)] * [PAR(BC,SUB)]; !
2216 6! CALL SAERO ( , BC, MINDEX, SUB, SYM, QDP, STABCF, !
2217 6! BGPDT(BC), [LHSA(BC,SUB)], [RHSA(BC,SUB)], [AAR], !
2218 6! [DELTA(SUB)], [PRIGID], [R33] ); !
2219 6! [AAL] := [D(BC)] * [AAR]; !
2220 6! CALL ROWMERGE ( [AAA(SUB)], [AAR], [AAL], [PARL(BC)] ); !
2221 6! [UAR] := [K1112(BC,SUB)] * [AAR] + [PAR(BC,SUB)] * !
2222 6! [DELTA(SUB)]; !
2223 6! [UAL] := [R1112(BC,SUB)] * [UAR] + [R1113(BC,SUB)] * [AAR] !
2224 6! - [R11PAL(BC,SUB)] * [DELTA(SUB)]; !
2225 6! CALL ROWMERGE ( [UAA(SUB)], [UAR], [UAL], [PARL(BC)] ); !
2226 6! IF NOMIT <> 0 [PAO(SUB)] := [POARO(BC,SUB)] * [DELTA(SUB)]; !
2227 6! ELSE !
2228 6!$ $!

```

Standard MAPOL Sequence Listing — Continued

```

STAT LEVL
2229 6:$ NO SUPPORT SET REDUCTION $!
2230 6:$ $!
2231 6:$ $!
2232 6:$ PROCESS STEADY AEROELASTIC DISCIPLINE $!
2233 6:$ $!
2234 6: PRINT("LOG=(' >>>DISCIPLINE: STEADY AERO')"); !
2235 6: ENDF; !
2236 5: ENDDO; !
2237 4: ENDF; !
2238 3:$ $!
2239 3:$ PERFORM ANY DYNAMIC ANALYSES -- NOTE THAT THESE ARE INDEPENDENT $!
2240 3:$ OF THE SUPPORT SET $!
2241 3:$ $!
2242 3: IF BDDYN <> 0 THEN !
2243 4: IF BFLUTR <> 0 THEN !
2244 5: PRINT("LOG=(' >>>DISCIPLINE: FLUTTER')"); !
2245 5: SUB := 0; !
2246 5: LOOP := TRUE; !
2247 5: WHILE LOOP DO !
2248 6: SUB := SUB + 1; !
2249 6: CALL FLUTDRV ( BC, SUB, LOOP ); !
2250 6: CALL FLUTQHHL ( , BC, SUB, ESIZE(BC), PSIZE(BC), [QKKL], !
2251 6: [UGTKA], [PHIA], USET(BC), !
2252 6: [TMN(BC)], [GSUBO(BC)], NGDR, AECOMPU, GEOMUA, !
2253 6: [PHIKH], [QHHLFL(BC,SUB)], OAGRDDSP ); !
2254 6: CALL FLUTDMA ( , BC, SUB, ESIZE(BC), PSIZE(BC), BGPDT(BC), !
2255 6: USET(BC), [MAA], [KAA], [TMN(BC)], [GSUBO(BC)], !
2256 6: NGDR, LAMBDA, [PHIA], [MHHFL(BC,SUB)], !
2257 6: [BHHFL(BC,SUB)], [KHHFL(BC,SUB)] ); !
2258 6: CALL FLUTTRAN ( , BC, SUB, [QHHLFL(BC,SUB)], LAMBDA, HSIZE(BC), !
2259 6: ESIZE(BC), [MHHFL(BC,SUB)], [BHHFL(BC,SUB)], !
2260 6: [KHHFL(BC,SUB)], CLAMBDA ); !
2261 6: ENDDO; !
2262 5: ENDF; !
2263 4: IF BDRSP <> 0 THEN !
2264 5: IF BMTR <> 0 OR BDTR <> 0 THEN !
2265 6: PRINT("LOG=(' >>>DISCIPLINE: TRANSIENT RESPONSE')"); !
2266 6: ENDF; !
2267 5: IF BMFR <> 0 OR BDFR <> 0 THEN !
2268 6: PRINT("LOG=(' >>>DISCIPLINE: FREQUENCY RESPONSE')"); !
2269 6: ENDF; !
2270 5: CALL QHHLGEN (BC, ESIZE(BC), [QKKL], [QKJL], [UGTKA], [PHIA], !
2271 5: [PHIKH], [QHHL], [QHJL]); !
2272 5: CALL DMA ( , BC, ESIZE(BC), PSIZE(BC), BGPDT(BC), USET(BC), [MAA], !
2273 5: [KAA], [TMN(BC)], [GSUBO(BC)], NGDR, !
2274 5: LAMBDA, [PHIA], [MDD], [BDD], [KDDT], [KDDF], !
2275 5: [MHH], [BHH], [KHHT], [KHHF] ); !
2276 5: CALL DYNLOAD ( , BC, GSIZE, ESIZE(BC), PSIZE(BC), SMPLOD, BGPDT(BC), !
2277 5: USET(BC), [TMN(BC)], [GSUBO(BC)], !
2278 5: NGDR, [PHIA], [QHJL], [PDT], [PDF], !
2279 5: [PTGLOAD], [PTHLOAD], [PFGLOAD], [PFHLOAD] ); !
2280 5: CALL DYNRSP (BC, ESIZE(BC), [MDD], [BDD], [KDDT], [KDDF], !

```

Standard MAPOL Sequence Listing — Continued

```

STAT LEVL
2281 5!           {MHH}, {BHH}, {KHHT}, {KHHF}, {PDT}, {PDF},           !
2282 5!           {QHHL}, {UTRANA}, {UFREQA}, {UTRANI}, {UFREQI},       !
2283 5!           {UTRANE}, {UFREQE} );                               !
2284 5!           IF BMTR <> 0 {UTRANA} := {PHIA} * {UTRANI};           !
2285 5!           IF BMFR <> 0 {UFREQA} := {PHIA} * {UFREQI};           !
2286 5!           ENDIF;                                              !
2287 4!           ENDIF;                                              !
2288 3!           IF BBLAST <> 0 THEN                                     !
2289 4!             PRINT("LOG=('          >>>DISCIPLINE: BLAST')");     !
2290 4!             CALL BLASTFIT ( BC, {QJLJ}, {MATTR}, {MATSS}, {BQDP}, {BFRC}, !
2291 4!               {DWNWSH}, HSIZE(BC), {ID2}, {MPART}, {UGTKA},       !
2292 4!               {BLGTJA}, {BLSTJA} );                               !
2293 4!             CALL COLPART ( {PHIA}, , {PHIE}, {MPART} );           !
2294 4!             CALL ROWMERGE ( {PHIR}, {ID2}, {D(BC)}, {PARL(BC)} );   !
2295 4!             CALL COLMERGE ( {PHIB}, {PHIR}, {PHIE}, {MPART} );     !
2296 4!             {GENM} := TRANS ( {PHIB} ) * [ {MAA} * {PHIB} ];       !
2297 4!             {GENK} := TRANS ( {PHIB} ) * [ {KAA} * {PHIB} ];       !
2298 4!             {DTSLP} := TRANS ( {BLSTJA} ) * {PHIB};               !
2299 4!             {FTF} := TRANS ( {PHIB} ) * {BLGTJA};                 !
2300 4!             {GENF} := {BQDP} {FTF} * {BFRC};                     !
2301 4!             {GENFA} := {BQDP} {FTF} * {MATSS};                     !
2302 4!             {GENQ} := {GENFA} * {DTSLP};                           !
2303 4!             {GENQL} := {BQDP} {FTF} * {MATTR};                     !
2304 4!             CALL PARTN ( {GENQ}, {QRR}, , {QRE}, {QEE}, {MPART} );   !
2305 4!             CALL PARTN ( {GENK}, , , {KEE}, {MPART} );             !
2306 4!             {KEQE} := {QEE} + {KEE};                               !
2307 4!             CALL DECCMP ( {KEQE}, {LKQ}, {UKQ} );                 !
2308 4!             CALL ROWPART ( {GENF}, {GFR}, {GFE}, {MPART} );         !
2309 4!             CALL GFBS ( {LKQ}, {UKQ}, {GFE}, {BTEM} );             !
2310 4!             {DELM} := -{QRE} * {BTEM} + {GFR};                     !
2311 4!             CALL BLASTTRIM ( BC, {DELM}, {MRR(BC)}, {URDB}, {DELB} ); !
2312 4!             {ELAS} := {BTEM} * {DELB};                             !
2313 4!             {SLPMOD} := TRANS ( {BLSTJA} ) * {PHIE};               !
2314 4!             CALL BLASTDRV ( BC, {GENM}, {GENK}, {GENFA}, {GENQL}, {DELB}, !
2315 4!               {URDB}, {DWNWSH}, {SLPMOD}, {ELAS}, {UBLASTI} );       !
2316 4!           ENDIF;                                              !
2317 3!$          $!
2318 3!$          BEGIN THE DATA RECOVERY OPERATIONS                 $!
2319 3!$          $!
2320 3!           IF NBNDCOND > 1 CALL NULLMAT ( {UF}, {AF}, {PHIF} );     !
2321 3!           IF NGDR <> 0 THEN                                         !
2322 4!$          $!
2323 4!$          DATA RECOVERY WITH GDR                                $!
2324 4!$          APPEND THE GDR-GENERATED DOFS TO THE F-SET              $!
2325 4!$          $!
2326 4!           PRINT("LOG=('          DYNAMIC REDUCTION RECOVERY')");   !
2327 4!           IF BLOAD <> 0 THEN                                         !
2328 5!             {UFGDR} := {GSUBO(BC)} * {UA};                         !
2329 5!             CALL ROWPART ( {UA}, {JJK}, , {PAJK} );                 !
2330 5!             CALL ROWMERGE ( {UF}, {UJK}, {UFGDR}, {PFJK} );         !
2331 5!             IF NRSET <> 0 THEN                                         !
2332 6!               {AFGDR} := {GSUBO(BC)} * {AA};                         !

```

Standard MAPOL Sequence Listing — Continued

```

STAT LEVL
2333 6!          CALL ROWPART ( [AA], [UJK], , [PAJK] );
2334 6!          CALL ROWMERGE ( [AF], [UJK], [AFGDR], [PFJK] );
2335 6!          ENDIF;
2336 5!          ENDIF;
2337 4!          IF BSAERO <> 0 THEN
2338 5!              FOR S = 1 TO SUB DO
2339 6!                  [UFGDR] := [GSUBO(BC)] * [UAA(S)];
2340 6!                  CALL ROWPART ( [UAA(S)], [UJK], , [PAJK] );
2341 6!                  CALL ROWMERGE ( [UAFTMP], [UJK], [UFGDR], [PFJK] );
2342 6!$
2343 6!$          MERGE THE CURRENT SUBCASE DEPENDENT RESULTS INTO A SINGLE
2344 6!$          MATRIX OF RESPONSE QUANTITIES FOR FURTHER RECOVERY
2345 6!$
2346 6!          CALL SAEROMRG ( BC, S, [UAF], [UAFTMP] );
2347 6!          IF NRSET <> 0 THEN
2348 7!              [AFGDR] := [GSUBO(BC)] * [AAA(S)];
2349 7!              CALL ROWPART ( [AAA(S)], [UJK], , [PAJK] );
2350 7!              CALL ROWMERGE ( [AAFTMP], [UJK], [AFGDR], [PFJK] );
2351 7!              CALL SAEROMRG ( BC, S, [AAF], [AAFTMP] );
2352 7!          ENDIF;
2353 6!          ENDDO;
2354 5!          ENDIF;
2355 4!          IF BMODES <> 0 THEN
2356 5!              [UFGDR] := [GSUBO(BC)] * [PHIA];
2357 5!              CALL ROWPART ( [PHIA], [UJK], , [PAJK] );
2358 5!              CALL ROWMERGE ( [PHIF], [UJK], [UFGDR], [PFJK] );
2359 5!          ENDIF;
2360 4!          IF BDTR <> 0 OR BMTR <> 0 THEN
2361 5!              [UFGDR] := [GSUBO(BC)] * [UTRANA];
2362 5!              CALL ROWPART ( [UTRANA], [UJK], , [PAJK] );
2363 5!              CALL ROWMERGE ( [UTRANF], [UJK], [UFGDR], [PFJK] );
2364 5!          ENDIF;
2365 4!          IF BDFR <> 0 OR BMFR <> 0 THEN
2366 5!              [UFGDR] := [GSUBO(BC)] * [UFREQA];
2367 5!              CALL ROWPART ( [UFREQA], [UJK], , [PAJK] );
2368 5!              CALL ROWMERGE ( [UFREQF], [UJK], [UFGDR], [PFJK] );
2369 5!          ENDIF;
2370 4!          ELSE
2371 4!              IF NOMIT <> 0 THEN
2372 5!$
2373 5!$          DATA RECOVERY WITH STATIC CONDENSATION
2374 5!$
2375 5!          PRINT("LOG=('          STATIC CONDENSATION RECOVERY')");
2376 5!          IF BLOAD <> 0 THEN
2377 6!              CALL RECOVA ( [UA], [PO], [GSUBO(BC)], NRSET, [AA],
2378 6!                  [IFM(BC)], , [KOOINV(BC)], [PFOA(BC)], [UF] );
2379 6!              IF NRSET <> 0 CALL RECOVA ( [AA], , [GSUBO(BC)], , , ,
2380 7!                  [PFOA(BC)], [AF] );
2381 6!          ENDIF;
2382 5!          IF BSAERO <> 0 THEN
2383 6!              FOR S = 1 TO SUB DO
2384 7!                  CALL RECOVA ( [UAA(S)], [PAO(S)], [GASUBO(BC,S)],

```

Standard MAPOL Sequence Listing — Continued

```

STAT LEVL
2385 7! NRSET, [AAA(S)], [IFMA(BC,S)], BSAERO, !
2386 7! [KOOL(BC,S)], [KOOU(BC,S)], !
2387 7! [PFOA(BC)], [UAFTMP] ); !
2388 7!$ !
2389 7!$ MERGE THE CURRENT SUBCASE DEPENDENT RESULTS INTO A SINGLE $!
2390 7!$ MATRIX OF RESPONSE QUANTITIES FOR FURTHER RECOVERY $!
2391 7!$ !
2392 7! CALL SAEROMRG ( BC, S, [UAF], [UAFTMP] ); !
2393 7! IF NRSET <> 0 THEN !
2394 8! CALL RECOVA ( [AAA(S)], [GASUBO(BC,S)], !
2395 8! [PFOA(BC)], [AAFTMP] ); !
2396 8! CALL SAEROMRG ( BC, S, [AAF], [AAFTMP] ); !
2397 8! ENDIF; !
2398 7! ENDDO; !
2399 6! ENDIF; !
2400 5! IF BMODES <> 0 THEN !
2401 6! [PHIO] := [GASUBO(BC)] * [PHIA]; !
2402 6! CALL ROWMERGE ( [PHIF], [PHIO], [PHIA], [PFOA(BC)] ); !
2403 6! ENDIF; !
2404 5! IF BDTR <> 0 OR BMTR <> 0 THEN !
2405 6! CALL RECOVA ( [UTRANA], [GASUBO(BC)], !
2406 6! [PFOA(BC)], [UTRANF] ); !
2407 6! ENDIF; !
2408 5! IF BDFR <> 0 OR BMFR <> 0 THEN !
2409 6! CALL RECOVA ( [UFREQA], [GASUBO(BC)], !
2410 6! [PFOA(BC)], [UFREQF] ); !
2411 6! ENDIF; !
2412 5! ELSE !
2413 5!$ $!
2414 5!$ DATA RECOVERY WITHOUT F-SET REDUCTION $!
2415 5!$ $!
2416 5! IF BLOAD <> 0 THEN !
2417 6! [UF] := [UA]; !
2418 6! IF NRSET <> 0 [AF] := [AA]; !
2419 6! ENDIF; !
2420 5! IF BSAERO <> 0 THEN !
2421 6! FOR S = 1 TO SUB DO !
2422 7!$ $!
2423 7!$ MERGE THE CURRENT SUBCASE DEPENDENT RESULTS INTO A SINGLE $!
2424 7!$ MATRIX OF RESPONSE QUANTITIES FOR FURTHER RECOVERY $!
2425 7!$ !
2426 7! CALL SAEROMRG ( BC, S, [UAF], [UAA(S)] ); !
2427 7! IF NRSET <> 0 CALL SAEROMRG ( BC, S, [AAF], [AAA(S)] ); !
2428 7! ENDDO; !
2429 6! ENDIF; !
2430 5! IF BMODES <> 0 [PHIF] := [PHIA]; !
2431 5! IF BDTR <> 0 OR BMTR <> 0 [UTRANF] := [UTRANA]; !
2432 5! IF BDFR <> 0 OR BMFR <> 0 [UFREQF] := [UFREQA]; !
2433 5! ENDIF; !
2434 4! ENDIF; !
2435 3!$ $!
2436 3! IF NBNDCOND > 1 CALL NULLMAT ( [UN], [AN], [PHIN] ); !

```

Standard MAPOL Sequence Listing — Continued

```

STAT LEVL
2437 3! IF NSPC <> 0 THEN
2438 4!$
2439 4!$ DATA RECOVERY WITH SPC-REDUCTION
2440 4!$
2441 4! PRINT('LOG=(' SPC RECOVERY')');
2442 4! IF BLOAD <> 0 THEN
2443 5! CALL YSMERGE ( [UN], [YS(BC)], [UF], [PNSF(BC)] );
2444 5! CALL OFFSPCF ( 0, BC, 1, 1, GSIZE, ESIZE(BC), NGDR,
2445 5! [KFS], [KSS], [UF], [YS(BC)], [PS],
2446 5! [PNSF(BC)], [PGMN(BC)], [PFJK], , , ,
2447 5! BGPDT(BC), OGRIDLOD );
2448 5! IF NRSET <> 0 CALL YSMERGE ( [AN], , [AF], [PNSF(BC)] );
2449 5! ENDIF;
2450 4! IF BSAERO <> 0 THEN
2451 5! CALL YSMERGE ( [UAN], [YS(BC)], [UAF], [PNSF(BC)] );
2452 5! IF NRSET <> 0 CALL YSMERGE ( [AAN], , [AAF], [PNSF(BC)] );
2453 5! ENDIF;
2454 4! IF BMODES <> 0 THEN
2455 5! CALL YSMERGE ( [PHIN], [YS(BC)], [PHIF],
2456 5! [PNSF(BC)] );
2457 5! IF DMODES <> 0 CALL OFFSPCF ( 0, BC, 2, 1, GSIZE,
2458 6! ESIZE(BC), NGDR,
2459 6! [KFS], , [PHIF], , ,
2460 6! [PNSF(BC)], [PGMN(BC)], [PFJK],
2461 6! , , , BGPDT(BC), OGRIDLOD );
2462 5! ENDIF;
2463 4! IF BDTR <> 0 OR BMTR <> 0
2464 5! CALL YSMERGE ( [UTRANN], [YS(BC)], [UTRANF],
2465 5! [PNSF(BC)], BDTR );
2466 4! IF BDFR <> 0 OR BMFR <> 0
2467 5! CALL YSMERGE ( [UFREQN], [YS(BC)], [UFREQF],
2468 5! [PNSF(BC)], BDFR );
2469 4! IF BFLUTR <> 0
2470 5! CALL OFFSPCF ( 0, BC, 4, 2, GSIZE, ESIZE(BC), NGDR, [KFS], ,
2471 5! [PHIF], , , [PNSF(BC)], [PGMN(BC)], [PFJK],
2472 5! , , , BGPDT(BC), OGRIDLOD );
2473 4! IF BBLAST <> 0 THEN
2474 5! [UBLASTF] := [PHIF]*[UBLASTI];
2475 5! CALL OFFSPCF ( 0, BC, 8, 1, GSIZE, ESIZE(BC), NGDR,
2476 5! [KFS], , [UBLASTF], , , [PNSF(BC)], [PGMN(BC)],
2477 5! [PFJK], , , , BGPDT(BC), OGRIDLOD );
2478 5! ENDIF;
2479 4! ELSE
2480 4!$
2481 4!$ DATA RECOVERY WITHOUT SPC-REDUCTION
2482 4!$
2483 4! IF BLOAD <> 0 THEN
2484 5! [UN] := [UF];
2485 5! IF NRSET <> 0 [AN] := [AF];
2486 5! ENDIF;
2487 4! IF BSAERO <> 0 THEN
2488 5! [UAN] := [UAF];

```

Standard MAPOL Sequence Listing — Continued

```

STAT LEVL
2489 5!          IF NRSET <> 0 [AAN] := [AAF];          !
2490 5!          ENDIF;                                !
2491 4!          IF BMODES <> 0 [PHIN] := [PHIF];        !
2492 4!          IF BDTR <> 0 OR BMTR <> 0 [UTRANN] := [UTRANA]; !
2493 4!          IF BDFR <> 0 OR BMFR <> 0 [UFREQN] := [UFREQA]; !
2494 4!          ENDIF;                                !
2495 3!$         IF NBNDCOND > 1 CALL NULLMAT ( [UG(BC)], [AG(BC)], [UAG(BC)], [AAG(BC)], !
2496 3!           [PHIG(BC)] );                          !
2497 4!          IF NMPC <> 0 THEN                        !
2498 3!          DATA RECOVERY WITH MPC-REDUCTION      !
2499 4!$         PRINT("LOG=('          MPC RECOVERY')"); !
2500 4!$         IF BLOAD <> 0 THEN                      !
2501 4!$         [UM] := [TMN(BC)] * [UN];              !
2502 4!           CALL ROWMERGE ( [UG(BC)], [UM], [UN], [PGMN(BC)] ); !
2503 5!           IF NRSET <> 0 THEN                      !
2504 5!             [UM] := [TMN(BC)] * [AN];              !
2505 5!             CALL ROWMERGE ( [AG(BC)], [UM], [AN], [PGMN(BC)] ); !
2506 5!           ENDIF;                                !
2507 6!           IF BSAERO <> 0 THEN                    !
2508 6!             [UM] := [TMN(BC)] * [UAN];              !
2509 6!             CALL ROWMERGE ( [AAG(BC)], [UM], [UAN], [PGMN(BC)] ); !
2510 6!           ENDIF;                                !
2511 5!           ENDIF;                                !
2512 4!           IF BMODES <> 0 THEN                    !
2513 4!             [UM] := [TMN(BC)] * [PHIN];            !
2514 4!             CALL ROWMERGE ( [PHIG(BC)], [UM], [PHIN], [PGMN(BC)] ); !
2515 4!           ENDIF;                                !
2516 5!           IF BDTR <> 0 OR BMTR <> 0 THEN          !
2517 5!             [UM] := [TMN(BC)] * [UTRANN];          !
2518 5!             CALL ROWMERGE ( [UTRANG], [UM], [UTRANN], [PGMN(BC)] ); !
2519 5!           ENDIF;                                !
2520 4!           IF BDFR <> 0 OR BMFR <> 0 THEN          !
2521 4!             [UM] := [TMN(BC)] * [UFREQN];          !
2522 4!             CALL ROWMERGE ( [UFREQG], [UM], [UFREQN], [PGMN(BC)] ); !
2523 4!           ENDIF;                                !
2524 4!           ELSE                                    !
2525 4!$          DATA RECOVERY WITHOUT MPC-REDUCTION    !
2526 4!$          IF BLOAD <> 0 THEN                      !
2527 4!$            [UG(BC)] := [UN];                      !
2528 4!$            IF NRSET <> 0 [AG(BC)] := [AN];        !
2529 4!$          ENDIF;                                !
2530 4!$          IF BSAERO <> 0 THEN                    !
2531 4!$            [UAG(BC)] := [UAN];                    !

```

Standard MAPOL Sequence Listing — Continued

```

STAT DEVL
2541 5!      IF NRSET <> 0 [AA(BC)] := [AAN];
2542 5!      ENDIF;
2543 4!      IF BMODES <> 0 [PHIG(BC)] := [PHIN];
2544 4!      IF BDTR <> 0 OR BMTR <> 0 [UTRANG] := [UTRANN];
2545 4!      IF BDFR <> 0 OR BMFR <> 0 [UFREQG] := [UFREQN];
2546 4!      ENDIF;
2547 3!$
2548 3!$ RECOVER PHYSICAL BLAST DISCIPLINE DISPLACEMENTS
2549 3!$
2550 3!      IF BBLAST <> 0 [UBLASTG] := [PHIG(BC)] * [UBLASTI];
2551 3!$
2552 3!$ HANDLE OUTPUT REQUESTS
2553 3!$
2554 3!      PRINT("LOG=(          OUTPUT PROCESSING)");
2555 3!      IF BSAERO <> 0 THEN
2556 4!$
2557 4!$ RECOVER STATIC AEROELASTIC LOADS DATA
2558 4!$
2559 4!      LOOP := TRUE;
2560 4!      SUB := 0;
2561 4!      WHILE LOOP DO
2562 5!          SUB := SUB + 1;
2563 5!          CALL SAERODRV (BC, SUB, LOOP, MINDEX, SYM, MACH, QDP);
2564 5!$
2565 5!$ CALL THE TRIMMED LOADS COMPUTATION WITH PROPER MATRICES
2566 5!$
2567 5!      IF SYM = 1 THEN
2568 6!          CALL OFFPALOAD ( , BC, MINDEX, SUB, GSIZE, BGPDT(BC),
2569 6!              [GTKG], [GSTKG], QDP, [AIRFRG(MINDEX)],
2570 6!              [DELTA(SUB)], [AICMAT(MINDEX)],
2571 6!              [UAG(BC)], [MGG], [AAG(BC)], [KFS],
2572 6!              [KSS], [UAF], [YS(BC)], [PNSF(BC)],
2573 6!              [PGMN(BC)], [PFJK], NGDR, USET(BC),
2574 6!              OGRIDLOD);
2575 6!      ELSE
2576 6!          IF SYM = -1 THEN
2577 7!              CALL OFFPALOAD ( , BC, MINDEX, SUB, GSIZE, BGPDT(BC),
2578 7!                  [GTKG], [GSTKG], QDP, [AIRFRG(MINDEX)],
2579 7!                  [DELTA(SUB)], [AICMAT(MINDEX)],
2580 7!                  [UAG(BC)], [MGG], [AAG(BC)], [KFS],
2581 7!                  [KSS], [UAF], [YS(BC)], [PNSF(BC)],
2582 7!                  [PGMN(BC)], [PFJK], NGDR, USET(BC),
2583 7!                  OGRIDLOD);
2584 7!          ENDIF;
2585 6!      ENDIF;
2586 5!$
2587 5!$ CALL TO COMPUTE THE TRIMMED LOADS/DISPLACEMENTS ON THE
2588 5!$ AERODYNAMIC MODEL
2589 5!$
2590 5!      IF SYM = 1 THEN
2591 6!          CALL OFFPAERM ( NITOK, BC, MINDEX, SUB, GSIZE, GEOMSA,
2592 6!              [GTKG], [GSTKG], QDP, [AIRFRG(MINDEX)],

```


Standard MAPOL Sequence Listing — Continued

```

STAT LEVL
2593 6!                                {DELTA(SUB)}, {AICMAT(MINDEX)},      !
2594 6!                                {UAG(BC)}, OAGRDLOD, OAGRDDSP );      !
2595 6!                                ELSE                                  !
2596 6!                                IF SYM = -1 THEN                      !
2597 7!                                CALL OFFPAEROM ( NITER, BC, MINDEX, SUB, GSIZE, GEOMSA,      !
2598 7!                                {GTKG}, {GSTKG}, QDP, {AIRFRM(MINDEX)},      !
2599 7!                                {DELTA(SUB)}, {AICMAT(MINDEX)},      !
2600 7!                                {UAG(BC)}, OAGRDLOD, OAGRDDSP );      !
2601 7!                                ENDIF;                                !
2602 6!                                ENDDO;                                !
2603 5!                                ENDDO;                                !
2604 4!                                ENDDO;                                !
2605 3!                                IF BDRSP <> 0 THEN                      !
2606 4!                                CALL OFFDLOAD ( , BC, BGPDT(BC), PSIZE(BC), ESIZE(BC), {PHIG(BC)},      !
2607 4!                                {PTGLOAD}, {PTHLOAD}, {PFGLOAD}, {PFHLOAD}, OGRIDLOD );      !
2608 4!                                IF BDTR <> 0 OR BMTR <> 0                      !
2609 5!                                CALL OFFSPCF ( 0, BC, 5, 1, GSIZE, ESIZE(BC),      !
2610 5!                                NGDR, {KFS}, , {UTRANF}, , ,      !
2611 5!                                {PNSF(BC)}, {PGMN(BC)}, {PFJK},      !
2612 5!                                {PHIG(BC)}, {PTGLOAD}, {PTHLOAD},      !
2613 5!                                BGPDT(BC), OGRIDLOD );      !
2614 4!                                IF BDFR <> 0 OR BMFR <> 0                      !
2615 5!                                CALL OFFSPCF ( 0, BC, 6, 2, GSIZE, ESIZE(BC),      !
2616 5!                                NGDR, {KFS}, , {UFREQF}, , ,      !
2617 5!                                {PNSF(BC)}, {PGMN(BC)}, {PFJK},      !
2618 5!                                {PHIG(BC)}, {PFGLOAD}, {PFHLOAD},      !
2619 5!                                BGPDT(BC), OGRIDLOD );      !
2620 4!                                ENDDO;                                !
2621 3!                                CALL OFFLOAD ( NUMOPTBC, BC, , GSIZE, BGPDT(BC), PSIZE(BC),      !
2622 3!                                {PG} );      !
2623 3!                                CALL OFFDISP( NUMOPTBC, BC, , GSIZE, BGPDT(BC), ESIZE(BC), PSIZE(BC),      !
2624 3!                                OGRIDDSP, {UG(BC)}, {AG(BC)}, {UAG(BC)}, {AAG(BC)},      !
2625 3!                                {UBLASTG}, , {UTRANG}, {UTRANE}, {UFREQG}, {UFREQE},      !
2626 3!                                LAMBDA, {PHIG(BC)} );      !
2627 3!                                CALL EDR ( NUMOPTBC, BC, , NDV, GSIZE, EOSUMMARY, EODISC,      !
2628 3!                                GLBDES, LCCLVAR, {PTRANS},      !
2629 3!                                {UG(BC)}, {UAG(BC)}, , {UTRANG}, {UFREQG}, {PHIG(BC)} );      !
2630 3!                                CALL OFFPEDR ( BC, HSIZE(BC) );      !
2631 3!                                ENDDO;                                !
2632 2!                                ENDDO;                                !
2633 1!                                ENDDO;                                !

```

3. THE SOLUTION CONTROL PACKET

The solution control packet provides the means by which the user selects the optimization and analysis tasks to be performed by the ASTROS system, their order of execution and the engineering data related to each. The solution control commands are analogous in purpose to the NASTRAN Case Control commands but they are very different in form and subtly different in interpretation. Understanding the differences between ASTROS and NASTRAN in the area of solution control is fundamental in understanding multidisciplinary optimization in the ASTROS system because the solution control command structure follows directly from the ASTROS capability to perform multidisciplinary analyses in a single run. It is **critical** that the user clearly understand the subtleties of solution control syntax and hierarchies. This section, therefore, augments the presentation of the solution control mechanics with a discussion of the design considerations that are embodied in the solution control commands. The detailed definition of all solution control commands follows at the end of the chapter.

In ASTROS, the solution control is very closely linked to the structure of the standard MAPOL sequence. It may be advantageous for the beginning user to read the standard MAPOL sequence discussion in the preceding section and to study the Theoretical Manual discussion of multidisciplinary optimization before reading the remainder of this section.

The solution control packet is initiated with the keyword **SOLUTION** which follows the **DEBUG** and **MAPOL** packets (if present) in the input data stream. The packet is terminated when the **BULK DATA** packet, or the end of the input stream, is encountered. The data are composed of solution control statements which can begin in any column and can extend over multiple physical records. Each statement is formed from a combination of keywords separated by blanks or commas as indicated in the detailed syntactical descriptions at the end of the chapter. Further, each command keyword can be abbreviated by the first four (or more) characters of the keyword. The solution control packet follows a prescribed hierarchy with the following levels:

INITIAL LEVEL (Level 1)
TYPE OF BOUNDARY CONDITION (Level 2)
BOUNDARY CONDITION(S) (Level 3)
DISCIPLINE(S) (Level 4)

Each of these levels is discussed in the following sections and compared and contrasted to their NASTRAN counterparts. In addition to these hierarchical commands, there are commands for output processing that can occur at several levels in the hierarchy. This section presents the available commands and output quantities, but the reader is referred to Chapter 5 of this document for the in-depth presentation of ASTROS output processing.

The hierarchical nature of solution control means that, if the user enters a command at one level in the hierarchy, it remains in effect at *all* subsequent levels *at* or *below* the current one unless overridden. If it is overridden at the *same* level, that overwrites the original command. If, on the other hand, the command is overridden at a *lower* level, it only supercedes the original command for the duration of that level and lower levels. Solution Control reverts to use the higher level default after the lower level has been left. Table 14 describes how the commands move from one level to the next and the defaults that they use in each.

Table 14. Levels of Solution Control

CURRENT LEVEL IS:	INCREASING LEVELS			DECREASING LEVELS		
	IF COMMAND IS:	USE DEFAULTS FROM:	THEN MOVE TO:	IF COMMAND IS:	USE DEFAULTS FROM:	THEN MOVE TO:
LEVEL 1 (Initial)	ANALYZE OPTIMIZE	LEVEL 1	LEVEL 2			
LEVEL 2	BOUNDARY	LEVEL 2	LEVEL 3			
LEVEL 3	Discipline commands (e.g. STATICS)	LEVEL 3	LEVEL 4			
LEVEL 4				Discipline commands (e.g. STATICS)	LEVEL 3	LEVEL 4
				BOUNDARY	LEVEL 2	LEVEL 3
				END	LEVEL 1	LEVEL 1

The user must be aware of these hierarchies especially when requesting output at higher levels. It is possible to get print requests by *default* where they are not expected if one is not careful with the solution control hierarchy. Another common problem is to place an output request on the wrong side of a level-incrementing solution command thus placing a command at a higher level than expected. Consider the following two examples:

EXAMPLE 1	EXAMPLE 2
<pre> OPTIMIZE BOUNDARY SPC=1 LABEL = CASE 1 STATICS (MECH=10) ... LABEL = CASE 2 STATICS (MECH=20) ... LABEL = CASE 3 STATICS (MECH=30) ... END </pre>	<pre> OPTIMIZE BOUNDARY SPC=1 STATICS (MECH=10) LABEL = CASE 1 ... STATICS (MECH=20) LABEL = CASE 2 ... STATICS (MECH=30) LABEL = CASE 3 ... END </pre>

In example 1, there are three discipline commands, **STATICS**, and three **LABEL** commands, one for each discipline. The indenture in the example helps to explain the results of these commands. The first **STATICS** case will be labelled **CASE 2**, because the **LABEL** command appears at LEVEL 4 with the **STATICS (MECH=10)** command. Similarly, the second **STATICS** case will be labelled **CASE 3**. Finally, the third **STATICS** case will be labelled **CASE 1** because that particular **LABEL** command appeared at LEVEL 3 prior to **STATICS (MECH=10)**. Example 2 illustrates the probable intent of the user. Here, the **LABEL** commands are placed below the **STATICS** command. As a result, the **LABELs** match the cases.

3.1. TYPE OF BOUNDARY CONDITION

ASTROS has been designed primarily to be an automated design tool, but it can also perform analyses without doing any design. This is reflected in the division of the solution control packet into two subpackets, either of which is optional. The first, or **OPTIMIZE**, subpacket defines the boundary condition(s) and discipline(s) which will generate design constraints to be used in the redesign task. In defining an optimization boundary condition, the user either implicitly or explicitly specifies that constraints be applied to certain (discipline dependent) response quantities. ASTROS then considers the complete set of constraints from all disciplines in all optimization boundary conditions in the redesign task. The second, **ANALYZE**, subpacket defines analyses that are to be performed on the possibly redesigned structure. The **ANALYZE** subpacket is intended to provide the designer with the means to obtain additional output that is not desired during the optimization phase or to perform additional analyses which were not performed in the design task. It can also be used to perform analyses on structures that are not to be designed at all. The form of the solution control packet is then:

```

SOLUTION
  OPTIMIZE
    ...
    ...   Optimization Subpacket
    ...
  END
  ANALYZE
    ...
    ...   Analysis Subpacket
    ...
END

```

If optimization is being performed, the **OPTIMIZE** subpacket must precede the **ANALYZE** subpacket. Any number of boundary conditions and/or disciplines can be performed in either subpacket.

3.2. BOUNDARY CONDITION

Each analysis discipline requires a set of physical boundary conditions and, in the case of unrestrained structures, a set of fictitious supports. These are defined in ASTROS in a manner very similar to that in NASTRAN; namely, through the definition of multipoint constraints (**MPC**), single point constraints (**SPC**) and support points (**SUPPORT**). Unlike NASTRAN, however, ASTROS requires a more rigorous definition of a boundary condition. The reason for this is that the user must ensure that the system matrices at each stage of matrix reduction up to the analysis set are uniquely defined by the boundary condition specification. At or below the analysis set, certain disciplines allow looping over families of direct matrix input, damping options, transfer functions, etc. For example, if the user intends to perform a normal modes analysis, a modal transient analysis and a modal flutter analysis in the same boundary condition, ASTROS requires that the modal representation of the system under analysis be the same for each discipline in the boundary condition. This requirement, which is necessary to efficiently perform multidisciplinary analysis, adds a number of additional parameters to the boundary condition definition beyond the **MPC**, **SPC** and **SUPPORT** definitions. They include definitions to perform matrix reductions (available in NASTRAN through Bulk Data but not always selectable in the Case Control Deck) as well as selection of additional point degrees of freedom. In NASTRAN, these data are either implicitly selected through the rigid format selection and/or bulk data or are a "discipline option" in the case control deck. While the boundary condition definition in ASTROS appears to be very complex, it is relatively simple if one realizes that the fundamental purpose of the **BOUNDARY** command is to uniquely specify the system level matrices and the matrix reductions that should be performed on them.

There is one level of *boundary condition* specification which is not treated in the **BOUNDARY** command. It deals with symmetry options which play a restrictive role in multi-disciplinary analysis, especially for aerodynamic disciplines. The symmetry options are often limited by the nature of the structural and/or aerodynamic models that are defined in the bulk data packet. For example, if the structural model is a half model only, the user cannot specify that asymmetric structural boundary conditions be analyzed. As a more common example, the user might want to perform an asymmetric aeroelastic analysis with a structural half model. Unfortunately, this is not possible in ASTROS. Whenever possible, the implicit (model-defined) boundary condition specifications that existed in the NASTRAN bulk data definitions and in the interface between bulk data and solution control have been replaced with solution control dependent options. There are, however, still limitations imposed through

the interactions between the model and the solution control on combining symmetric/antisymmetric and asymmetric boundary conditions within a single run. The ten boundary condition specifications in AS-TROS are shown in the following table:

OPTION	DESCRIPTION
SPC	Selects single point constraints defining DOF's with fixed or prescribed motion.
MPC	Selects multi-point constraints defining dependency relations among specific DOF's.
REDUCE	Defines the DOF's to be retained after a Guyan reduction.
SUPPORT	Defines DOF's to provide support conditions for free-free modal extraction, inertial relief and aeroelastic analyses.
METHOD	Specifies an EIGR bulk data entry which gives eigenvalue extraction data if an eigenanalysis is to be performed.
DYNRED	Invokes dynamic reduction.
INERTIA	Specifies a JSET bulk data set for dynamic reduction.
ESET	Specifies the extra point DOF's to be included in dynamic response analyses.
K2GG	Specifies the name of the direct stiffness matrix input in the structural set (g-set) to be included in ALL analyses.
M2GG	Specifies the name of the direct mass matrix input in the structural set (g-set) to be included in ALL analyses.

A boundary condition is defined by the BOUNDARY request and one or more of these further specifications, each of which points to bulk data entries.

As enumerated above, the specification of METHOD and ESET at this level in the hierarchy is in recognition of the fact that a number of the disciplines could require different sets of data for the associated items and it is desirable to group operations with one set of items together. This does, by definition, create a restriction that only one eigenanalysis and only one size of p-size matrices can be accommodated per boundary condition. Examples of boundary definitions are:

```
BOUNDARY SPC = 100
BOUNDARY MPC = 10, SPC = 100
BOUNDARY SPC = 10, MPC = 20, REDUCE = 30, SUPPORT = 40
BOUNDARY SPC = 10, REDUCE = 20, METHOD = 100
BOUNDARY SPC = 1, K2GG = STIFF, M2GG = MASS
BOUNDARY SPC = 4, DYNRED = 2, INERTIA = 4
```

Note that all desired specifications are listed and that their order of appearance is not important. At least one option is required.

Several boundary conditions may appear within a given subpacket. For example:

```
ANALYZE
  BOUNDARY SPC = 10
    STATICS (MECH=5)
    ...
  BOUNDARY SPC = 20, REDUCE = 30, METHOD = 1111
    STATICS (THERM=10)
    ...
  MODES
    ...
END
```

In this case, a **STATICS** analysis is performed using the first boundary condition followed by a **STATICS** and modes analysis for the second boundary condition. Note that unlike NASTRAN, the sets of points to be retained in the Guyan reduction and used for the support definition are selected.

The appearance of a **BOUNDARY** command leads to expensive matrix partitioning and decomposition operations. Therefore, some thought should be expended to avoid unnecessary computer resource use. For example, suppose an ASTROS execution was directed to perform static analyses with two boundaries: **SPC=10** and **SPC=20**, and a dynamic analysis with two boundaries: **SPC=10** and **SPC=100**. The direct solution sequence could be:

```
ANALYZE
  BOUNDARY SPC = 10
    STATICS (MECH=10)
  BOUNDARY SPC = 20
    STATICS (MECH=20)
    ...
  BOUNDARY SPC = 10, METHOD = 30
    MODES
    ...
  BOUNDARY SPC = 100, METHOD = 40
    MODES
    ...
END
```

This sequence would cause four separate partitionings of the system level matrices. On the other hand, the sequence:

```
ANALYZE
  BOUNDARY SPC = 10, METHOD=30
  STATICS (MECH=10)
  ...
  MODES
  ...
  BOUNDARY SPC = 20
  STATICS ((MECH=20)
  ...
  ...
  BOUNDARY SPC = 100, METHOD = 40
  MODES
  ...
  ...
END
```

eliminates one of the four partitioning operations.

3.3. DISCIPLINES

A number of types of analyses, or disciplines, can be performed during a given **ANALYZE** or **OPTIMIZE** boundary condition. In fact, it is this multidisciplinary capability that makes the ASTROS code viable in a preliminary design context. The preceding sections have already alluded to the fact that each of these disciplines has an associated set of commands:

```
ANALYZE
  BOUNDARY SPC = 30
  DISCIPLINE 1
  ...
  ...
  DISCIPLINE 2
  ...
  ...
END
```


A suite of eight disciplines are available in ASTROS as shown in Table 15. Of these options, **TRANSIENT**, **FREQUENCY**, **BLAST** and **NPSAERO** do not generate any design constraints and so are not useful in **OPTIMIZE** boundary conditions. Should the user wish to see output from these disciplines during the optimization, however, they are supported in the **OPTIMIZE** subpacket with the exception of **NPSAERO**. Since **NPSAERO** does not require a structural model of any kind, it is restricted to the **ANALYZE** subpacket. As discussed in Chapter 2, its solution is performed in the ASTROS preface phase.

The standard MAPOL sequence contains almost no restrictions on the combination of disciplines and subcases in a boundary condition. **SAERO** disciplines, for example, require multiple symmetry, Mach number and dynamic pressure dependent correction matrices. The standard algorithm automatically re-sorts the user's input subcases to solve the maximum number of right hand sides for a given aeroelastic correction matrix. The results are then returned to the user's order with no limitations imposed. Similarly, the flutter discipline loops over a set of direct dynamic input matrices to accommodate multiple closed loop systems using a single set of structural matrices. The only limits are those of symmetry discussed earlier in which the structural and aerodynamic symmetries should be the same for all subcases in a boundary condition and the restriction to a single transient and a single frequency response per boundary condition.

Table 15. Summary of ASTROS Disciplines

DISCIPLINE	DESCRIPTION	DISCIPLINE	DESCRIPTION
STATICS	Static structural analysis	TRANSIENT	Transient response analysis
MODES	Normal modes of vibration	FREQUENCY	Frequency response analysis
SAERO	Steady-state aeroelastic analysis	BLAST	Transient response to a nuclear blast.
FLUTTER	Aeroelastic stability analysis	NPSAERO	Nonplanar rigid static aerodynamic analysis.

3.3.1. DISCIPLINE OPTIONS

Each of the disciplines requires further options to completely define the execution process. These options point to set IDs in the bulk data packet that define engineering data. For example, the **STATICS** discipline requires that loads information be supplied. This is implemented in ASTROS by a parenthetical "phrase" attached to the **STATICS** discipline:

```
SOLUTION
  OPTIMIZE STRATEGY=FSD
      ...
      ...
  STATICS (MECH=10)
      ...
      ...
END
```

In this case, bulk data applied load entries with a set ID of 10 are used to construct a mechanical load vector in a **STATICS** analysis. In general, the discipline commands have the form:

```
<disc> <type> [( <option> = <n>, <option> = <n> )]
```

The discipline options that are available are:

OPTION	DESCRIPTION
MECHANICAL GRAVITY THERMAL	Specify load set IDs for the STATIC discipline.
TRIM	Specifies a TRIM bulk data entry which gives flight condition information for the SAERO and BLAST disciplines.
DCON DCONSTRAINT	Specifies the set IDs of constraint bulk data entries that apply for the given discipline.
STRESS STRESSCONSTRAINT	Specifies the set IDs of stress constraint bulk data entries that apply for the given STATICS or SAERO discipline.
STRAIN STRAINCONSTRAINT	Specifies the set IDs of strain constraint bulk data entries that apply for the given STATICS or SAERO discipline.
DLOAD	Specifies applied loads for the TRANSIENT and FREQUENCY disciplines.
TSTEP	Specifies the time step for the TRANSIENT and BLAST disciplines as well as for the discrete form of the GUST discipline.
FSTEP	Specifies the frequencies for the FREQUENCY and the harmonic form of the GUST discipline.
IC	Specifies the initial conditions that are to be used in the direct method for the TRANSIENT discipline.
FFT	Specifies that the Fast Fourier technique is to be used in the TRANSIENT or GUST disciplines.
BLCOND	Specifies parameters for the BLAST discipline.
FLCOND	Specifies parameters for the FLUTTER discipline.

OPTION	DESCRIPTION
CONTROL	Specifies the name of a control surface modifier matrix for flutter analysis.
GUST	Specifies that a gust analysis is to be performed for the accompanying transient or frequency discipline.
K2PP	Specifies an input stiffness matrix on the physical degrees of freedom for FREQUENCY , TRANSIENT and FLUTTER disciplines.
M2PP	Specifies an input mass matrix on the physical degrees of freedom for FREQUENCY , TRANSIENT and FLUTTER disciplines.
B2PP	Specifies an input damping matrix on the physical degrees of freedom for FREQUENCY , TRANSIENT and FLUTTER disciplines.
TFL	Specifies transfer functions that are to be included in FREQUENCY , TRANSIENT and FLUTTER disciplines.
DAMPING	Specifies structural or viscous damping to be used in FREQUENCY , TRANSIENT and FLUTTER disciplines.

The discipline types are:

OPTION	DESCRIPTION
DIRECT	Specifies that the direct method is to be used in the TRANSIENT or FREQUENCY disciplines.
MODAL	Specifies that the modal method is to be used in the TRANSIENT or FREQUENCY disciplines.
SYMMETRIC	Specifies that the SAERO or NPSAERO subcase is to use aerodynamics derived with symmetric conditions about the Y=0 plane.
ANTISYMMETRIC	Specifies that the SAERO or NPSAERO subcase is to use aerodynamics derived with antisymmetric conditions about the Y=0 plane.

Table 16 presents a matrix that defines options and types available for each of the disciplines. In addition, disciplines requiring particular boundary condition specifications are noted; for example, modal disciplines require a **METHOD** specification on the **BOUNDARY** command. The following subsections present each discipline in turn to more explicitly define the discipline options. Most importantly, these subsections present the definition of a "subcase" of the discipline as it is defined in the **ASTROS** system and present the response quantities that can be constrained in the optimization task.

3.3.2. STATICS Discipline Options

One or more of the **MECHANICAL**, **GRAVITY** or **THERMAL** load specifications must be called out as a discipline option for **STATICS**. Each **STATICS** discipline constitutes one subcase (one load vector) so specifying a combination of load types will generate a linear combination of the selected loads. A reference to the **LOAD** bulk data entry as a **MECHANICAL** load can also be used to obtain linear load combinations. If the **STATICS** discipline appears in the **OPTIMIZE** subpacket, the **DCONSTRAINT** option can be used to refer to **DCONDSP** bulk data entries to apply displacement constraints. Stress constraints defined on **DCONF**T, **DCONF**TM, **DCONF**TP, **DCONF**TW, **DCONF**TWM, **DCONF**TWP, **DCONF**VM, **DCONF**VMM, **DCONF**VMP are selected

Table 16. Summary of Discipline Options

COMMAND	DISCIPLINE							
	STAT	MODE	SAER	FLUT	TRAN	FREQ	BLAS	NPSA
MECH	○							
GRAV	○							
THERM	○							
TRIM			●					●
DCONSTRAINT	○	○	○	○				
STRESS	○		○					
STRAIN	○		○					
DLOAD					●	●		
TSTEP					●		●	
FSTEP						●		
IC					○			
FFT					○	○		
DIRECT					○	○		
MODAL					○	○		
BLCOND							●	
FLCOND				●	○			
GUST					○	○		
K2PP				○	○	○		
M2PP				○	○	○		
B2PP				○	○	○		
TFL				○	○	○		
DAMPING				○	○	○		
SYMMETRIC			○					○
ANTISYMMETRIC			○					○
Notes:								
Required Commands:		●						
Optional Commands:		○						

by the **STRESSCONSTRAINT** option. Strain constraints defined on **DCONEP**, **DCONEPM** and **DCONEPP** are selected by the **STRAINCONSTRAINT** option. All **DCONxxx** bulk data entries, such as **DCONTHK**, that do not have **SETID** fields will be applied to the model in combination with set selectable constraints to make up the set of design constraints.

3.3.3. MODES Discipline Options

MODES is completely defined for analysis by the **METHOD** boundary specification, which refers to an **EIGR** bulk data entry selecting the eigenvalue extraction method. If, however, the modal analysis is performed in the **OPTIMIZE** subpacket, the **DCONSTRAINT** option can be used to apply frequency constraints through the **DCONFRQ** bulk data entry. Note that more than one frequency can be constrained and that more than one constraint can be placed on the same modal frequency. The user is warned against defining the frequency constraints in such a way as to specify an excluded range of frequencies for a mode; for example, requiring that a modal frequency be below 10 Hz OR above 20 Hz. **ASTROS** treats all applied constraints as Boolean **AND** statements so the above example would be interpreted by **ASTROS** as an inconsistent requirement that the frequency be both above 20 Hz and below 10 Hz. All **DCONxxx** bulk data entries, such as **DCONTHK**, that do not have **SETID** fields will be applied to the model in combination with set selectable constraints to make up the set of design constraints.

In **ASTROS**, each eigenvector is considered to be a separate subcase. It is important to note in this case that more than one subcase is represented by a single solution control discipline statement. In output requests, therefore, the subcases for which output is desired must be explicitly selected. This is presented in greater detail in section 3.4 and in Chapter 5.

3.3.4. SAERO Discipline Options

The **SAERO** discipline must have a **TRIM** condition and symmetry type specified in the solution control. The symmetry default is **SYMMETRIC**. For analysis, this selection completes the specification of the discipline with each **TRIM** condition generating one subcase. In the **OPTIMIZE** subpacket, the **DCONSTRAINT** option can be used to select a number of different constraint types which depend on the type of **TRIM** analysis selected. In general the **DCONSTRAINT** can refer to **DCONDSP** bulk data entries for displacement constraints, **DCONCLA** for lift effectiveness constraints, **DCONALE** for aileron effectiveness constraints, **DCONSCF** for stability coefficient constraints and **DCONTRM** for constraints on trim parameters. The **SAERO** discipline *always* generates a static displacement field to which any static constraint may be applied. Stress constraints defined on **DCONFT**, **DCONFTM**, **DCONFTP**, **DCONTW**, **DCONTWM**, **DCONTWP**, **DCONVM**, **DCONVMM**, **DCONVMP** are selected by the **STRESSCONSTRAINT** option. Strain constraints defined on **DCONEP**, **DCONEPM** and **DCONEPP** are selected by the **STRAINCONSTRAINT** option. All **DCONxxx** bulk data entries, such as **DCONTHK**, that do not have **SETID** fields will be applied to the model in combination with set selectable constraints to make up the set of design constraints.

3.3.5. FLUTTER Discipline Options

The **FLUTTER** discipline must have a flight condition specified in the solution control through the **FLCOND** option. In addition, the **K2PP**, **B2PP**, **M2PP**, **TFL** and **DAMPING** options may be used with or without an **ESET** Boundary Condition option to impose a case-by-case set of additional inputs/degrees-of-freedom for modelling control systems, etc. For analysis, this selection completes the specification of the discipline with each **FLCOND** condition generating up to one "subcase" (consisting of up to one flutter

eigenvector) for each Mach number and density ratio if flutter occurs. In the **OPTIMIZE** subpacket, the **DCONSTRAINT** option can be used to select **DCONFLT** bulk data entries to place a required damping limit on each of the roots extracted in the flutter analysis. The actual flutter root and eigenvector cannot be obtained in the **OPTIMIZE** subpacket.

3.3.6. TRANSIENT Discipline Options

The **TRANSIENT** discipline must have time step and load information specified in the solution control through the **TSTEP** and **DLOAD** options. This discipline has no associated constraints and, while it is fully supported in the **OPTIMIZE** subpacket, it will not generate data for use in the re-design task. There are many additional options which can be selected in transient analysis. These are 1) initial conditions, which can be selected through the **IC** option for **DIRECT** transient analyses; 2) Fast Fourier Transform techniques, which are selected with the **FFT** option; and 3) discrete gust loads, which are applied using the **GUST** option. In each case, the solution control option points to a bulk data entry having the same name. In addition, the **K2PP**, **B2PP**, **M2PP**, **TFL** and **DAMPING** options may be used with or without an **ESET** Boundary Condition option to impose a case-by-case set of additional inputs/degrees-of-freedom for modelling control systems, etc.

In **ASTROS**, each time step for which output is saved is considered to be a separate subcase. It is important to note that, like the **MODES** discipline, more than one subcase is represented by a single solution control discipline statement. In output requests, therefore, the subcases for which output is desired must be explicitly selected. This is presented in greater detail in section 3.4 and in Chapter 5.

3.3.7. FREQUENCY Discipline Options

The **FREQUENCY** discipline is very similar to the **TRANSIENT** discipline presented in the preceding subsection. Frequency step and load information are specified in the solution control through the **FSTEP** and **DLOAD** options. This discipline has no associated constraints and, while it is fully supported in the **OPTIMIZE** subpacket, it will not generate data for use in the re-design task. There are two additional options which can be selected in frequency response analysis. These are 1) Fast Fourier Transform techniques, which are selected with the **FFT** option; and 2) harmonic gust loads, which are applied using the **GUST** option. In each case, the solution control option points to a bulk data entry having the same name. In addition, the **K2PP**, **B2PP**, **M2PP**, **TFL** and **DAMPING** options may be used with or without an **ESET** Boundary Condition option to impose a case-by-case set of additional inputs/degrees-of-freedom for modelling control systems, etc.

In **ASTROS**, each frequency step for which output is saved is considered to be a separate subcase. It is important to note that, like the **MODES** discipline, more than one subcase is represented by a single solution control discipline statement. In output requests, therefore, the subcases for which output is desired must be explicitly selected. This is presented in greater detail in Subsection 3.4 and in Chapter 5.

3.3.8. BLAST Discipline Options

The **BLAST** discipline's solution control is a limited subset of those in the **TRANSIENT** discipline. It must have time steps specified in the solution control through the **TSTEP** option but, unlike the normal transient response, **DLOAD** is not used to generate the load. Instead, the **BLCOND** option is used to select a blast condition defined on a **BLAST** bulk data entry. As with the other dynamic response disciplines, there

are no associated design constraints. These two options along with the analysis type (**DIRECT** or **MODAL**) completely specify the **BLAST** analysis. The **K2PP**, **B2PP**, **M2PP**, **TFL** and **DAMPING** options *are not available*.

In **ASTROS**, each time step for which output is saved is considered to be a separate subcase. It is important to note that, like the **MODES** discipline, more than one subcase is represented by a single solution control discipline statement. In output requests, therefore, the subcases for which output is desired must be explicitly selected. This is presented in greater detail in Subsection 3.4 and in Chapter 5.

3.3.9. **NPSAERO Discipline Options**

The **NPSAERO** discipline must have a **TRIM** condition and symmetry type specified in the solution control and is only available in the **ANALYZE** subpacket. The symmetry default is **SYMMETRIC**. This selection completes the specification of the discipline with each **TRIM** condition generating one subcase. The results are the rigid aerodynamic loads on the aerodynamic model. No structural model is needed and flexibility effects are not included in the analysis.

3.4. **OUTPUT REQUESTS**

Most analysis disciplines in **ASTROS** have response quantities (displacements, stresses, strains, etc.) computed at either grid points, structural elements or aerodynamic elements. The user can select that these results be written to the print (output) file through the **PRINT** command and its associated options or written to a punch file through the **PUNCH** command. In addition, there are a number of solution control commands that can be used to label the output. This subsection documents the **PRINT** and **PUNCH** commands and the labeling commands and discusses their use. The **PRINT** and **PUNCH** commands are identical in form and interpretation, so the **PRINT** command will be used to represent both commands in the following discussion. There are also many features and utilities available to the user to obtain output through modifications to the executive **MAPOL** sequence. These include direct use of **MAPOL** utilities, modification of print parameters in functional module calling sequences and user written procedures or modules. These output capabilities and a more complete discussion of the output processing (**PRINT** and **PUNCH**) capabilities of the **ASTROS** system is presented in Chapter 5 of this manual.

The **PRINT** and **PUNCH** commands have a number of options which can be separated into three groups: subset options, response quantity options and form options. The subset options select a set of subcases and/or design iterations to which the **PRINT** command applies while the remaining options select the actual data quantities that are desired; (e.g. stresses, strains, and displacements) and the form in which complex quantities are to be printed. The output selection can appear at any level of the solution control hierarchy and will apply at that level until it is overridden. When more than one discipline is covered by a print request at the boundary level, **ASTROS** will consider only the relevant print requests for each discipline. For example, if **STATICS** and flutter are performed, the **STATICS** discipline will ignore any **ROOTS** requests and the flutter discipline will ignore any **STRESS** requests.

3.4.1. Subset Options

As alluded to in the preceding subsections, some disciplines have more than one *subcase* per solution control statement. Others, like **STATICS** and **SAERO** have a separate solution control statement for each subcase. In all cases, disciplines within the **OPTIMIZE** subpacket may be analyzed at one or more design iterations. When one subcase is defined per statement, the user is free to modify the print requests from subcase to subcase; for example:

```
ANALYZE
  BOUNDARY SPC = 10
    STATICS ( MECH = 10 )
      PRINT STRESS = ALL, DISP = 100
    STATICS ( MECH = 20, GRAV = 100 )
      PRINT DISP = ALL
```

specifies that stresses for all elements and displacements for nodes listed in set 100 be printed for the first subcase (mechanical loads with set identification 10) and only the displacements be printed for the next load condition. When the discipline generates more than one subcase, however, the user *must* specify the subcases to which the **PRINT** request applies. For example:

```
ANALYZE
  BOUNDARY SPC=10, METHOD=1000
  MODES
  PRINT (MODES=ALL) DISP=100
```

selects the displacements (eigenvectors) for all the computed mode shapes be printed. If the **MODES=ALL** selection were not included in the **PRINT** statement, the user would get *no* output at all. The user is cautioned that the output processing in **ASTROS** is designed to limit output to those quantities that are explicitly selected and, therefore, the default for subcase option **MODES** is that no modes are selected. Whenever multiple subcases are generated by a discipline, as in the case of **MODES**, **TRANSIENT**, **FREQUENCY** and **BLAST**, a subcase selection option is *required* on the **PRINT** command in order to get any output.

If the discipline appears in the **OPTIMIZE** subpacket, the user may request that the output appear only at certain iterations. For example:

```
OPTIMIZE
  BOUNDARY SPC=10, METHOD=1000
  MODES (DCON=1000)
  PRINT (ITER=10, MODES=ALL) DISP=100
```

selects the displacements (eigenvectors) for all the computed mode shapes be printed at the iterations given in **ITERLIST 10**. Unlike the other subset selectors, the default for **ITER** is **ALL**. Omission of the **ITER** selector therefore implies that the quantity will be printed at every iteration. This default is a consequence of compatibility with early versions of **ASTROS** in which there was no **ITERATION** selection at all.

The subset selections can be specified at two levels as parenthetical *phrases* attached to the print or punch statement. At the higher level, the subset options generate defaults for the entire print or punch statement. For example:


```
PRINT (ITER=10, TIME=ALL) STRESS=ALL, STRAIN=ALL
```

requests that all stresses and strains at all time steps for the iterations in **ITERLIST** 10 be printed. In addition, the subset options can be attached to the individual quantity options to override the print default. For example:

```
PRINT (ITER=10, TIME=ALL) STRESS=ALL, STRAIN(TIME=10)=ALL
```

overrides the **TIME=ALL** default for the strain output. At both levels, the defaults are **NONE** for **TIME**, **FREQ** and **MODE** and **ALL** for **ITER**.

The subset options in **ASTROS** are:

OPTION	DESCRIPTION
FREQUENCY	Selects the frequency steps of frequency response disciplines at which output is desired by referencing a FREQLIST bulk data entry.
ITERATION	Selects the design iterations at which output is desired by referencing a ITERLIST bulk data entry.
MODE	Selects the eigenvectors of a normal modes discipline at which output is desired by referencing a MODELIST bulk data entry.
TIME	Selects the time steps of transient response and blast analyses at which output is desired by referencing a TIMELIST bulk data entry.

3.4.2. Response Quantity Options

ASTROS is able to compute a number of response quantities for each discipline type. Each discipline type generates a different set of quantities so that the quantity selected by a particular keyword can sometimes change from one discipline to another. In addition, the available quantities are a sometimes a function of the boundary condition type. For example, the flutter mode shape is not available as an output from a flutter analysis performed in the **OPTIMIZE** subpacket. This subsection will present the available quantities, the **PRINT** options which select them and the limitations (if any) on their availability.

The following table summarizes the available **PRINT** and **PUNCH** response quantity options:

OPTION	DESCRIPTION
ACCELERATION	Selects accelerations at nodal points.
AIRDISPLACEMENT	Selects displacements on aerodynamic boxes.
CGRADIENT	Selects gradients of active constraints.
DCONSTRAINT	Selects active constraints at each iteration.
DISPLACEMENTS	Selects displacements at nodal points.
ENERGY	Selects strain energy at structural elements.

OPTION	DESCRIPTION
FORCE	Selects element forces at structural elements.
GDESIGN	Selects global design variables.
GPFORCE	Selects grid point forces at nodal points.
GPWG	Selects print of grid point weight summary.
KSNS	Selects stiffness sensitivities at design variables.
LDESIGN	Selects local design variables.
LOAD	Selects applied loads at nodal points.
MASS	Selects mass matrix at nodal points.
MODEL	Selects Bulk Data at current design point. (PUNCH only)
MSNS	Selects mass sensitivities at design variables.
OGRADIENT	Selects gradient of the objective function.
QHH	Selects QHH generalized unsteady aerodynamic forces at modes.
QHJ	Selects QHJ generalized unsteady aerodynamic forces at modes.
ROOT	Selects flutter and normal modes roots (eigenvalues).
SPCFORCE	Selects forces of single point constraint at nodal points.
STIFFNESS	Selects stiffness matrix at nodal points.
STRAIN	Selects strains at structural elements.
STRESS	Selects stresses at structural elements.
TPRESSURE	Selects trim pressures at aerodynamic boxes.
TRIM	Selects trim and stability coefficients for steady aeroelastic analyses.
VELOCITY	Selects velocities at nodal points.

As in NASTRAN, stresses, strains and element forces are computed in the element coordinate system at predetermined or user selected points in the element. Nodal quantities are computed in the global coordinate system. CGRA, DCON, GDES, KSNS, MODEL, MSNS, OGRA and HIST are only applicable in the OPTIMIZE subpacket *above* the first BOUNDARY (since these requests transcend all analyses). The DISP option for flutter analyses is only applicable in the ANALYZE subpacket. Other options are available independent of the boundary condition type. Table 17 presents a matrix of response quantity options for each discipline type, showing the applicability of each option. Any requests for quantities that do not apply to the particular discipline will be ignored by the output processor without warning.

Table 17. Response Quantities by Discipline

OPTION	DESIGN	STAT	MODE	SAERO	FLUT	TRANS	FREQ	BLAS	NPSA
ACCEL		✓		✓		✓	✓	✓	
AIRDISP				✓					
CGRADIENT	✓								
DCONSTRAINT	✓								
DISP		✓	✓	✓	✓	✓	✓	✓	
ENERGY		✓	✓	✓		✓		✓	
FORCE		✓	✓	✓		✓		✓	
GDESIGN	✓								
GPFORCE		✓		✓					
GPWG		✓	✓	✓	✓	✓	✓	✓	
KSNS	✓								
LDESIGN	✓								
LOAD		✓		✓		✓	✓	✓	✓
MASS		✓	✓	✓	✓	✓	✓	✓	
MODEL	✓								
MSNS	✓								
OGRADIENT	✓								
QHH					✓		✓		
QHJ							✓		
ROOT			✓		✓				
SPFORCE		✓		✓					
STIFFNESS		✓	✓	✓	✓	✓	✓	✓	
STRAIN		✓	✓	✓		✓		✓	
STRESS		✓	✓	✓		✓		✓	
TPRESSURE				✓					✓
TRIM				✓				✓	
VELO						✓	✓	✓	

Most options can be **ALL**, **NONE** or an integer value which selects bulk data entry sets listing the items for which the response quantity is desired. For example, the **STRESS** option points to the **ELEMLIST** bulk data entity which lists the elements for which stresses are desired. The **NONE** option is used to override a default established through a print or punch request at a higher level in the hierarchy. The ASTROS output philosophy is similar to that of NASTRAN in that it is assumed that mistakes in the output requests should not terminate execution. If, for example, the requested structural element does not exist in the model, the output request will be ignored without any warning to the user. Other output request errors in ASTROS are treated in a similar manner, occasionally generating a warning message, but more typically resulting in no visible indication that the request was in error. Therefore the user can, in most cases, request output that does not apply to the discipline, for entities (nodes or elements) which do not exist and/or for subcases that are not defined without causing termination of the execution.

3.4.3. Form Options

For complex response quantities, the form option is provided to select either **RECTANGULAR** or **POLAR** form. Rectangular form gives the cartesian components of the quantity in the rectangular complex plane in which the first number represents the real component and the second number the imaginary component. Polar form gives the components in polar coordinates in which the first number represents the radial distance from the origin (the magnitude) and the second represents the angular displacement from the real coordinate axis (the phase angle). The phase angle is computed in degrees.

The form can be specified at two levels as parenthetical *phrases* attached to the print or punch statement. At the higher level, the form option generates a default for the entire print or punch statement. For example:

```
PRINT (POLAR) STRESS=ALL, STRAIN=ALL
```

requests that polar form be used for both stress and strain response quantities. In addition, the form option can be attached to the individual quantity options to override the print default. For example:

```
PRINT (POLAR) STRESS=ALL, STRAIN(RECT)=ALL
```

overrides the polar default for the strain output. At both levels, the default form is rectangular and any polar requests for real output quantities are ignored.

3.4.4. Labeling Options

Labeling of printed output is performed through the use of three optional commands identical in form to their NASTRAN counterparts:

OPTION	DESCRIPTION
TITLE	A title header that will appear as the first line on each page of output.
SUBTITLE	A secondary header that will appear on the second line of each page of output.
LABEL	A tertiary header that is typically used to identify subcase (discipline level) output.

Each of these commands can appear at any level in the solution control hierarchy and will be applied until superseded.

3.5. SOLUTION CONTROL COMMANDS

The ASTROS Solution Control Commands are described in this section.

Solution Control Command: \$

Description: Allows commentary data to be placed in the Solution Control packet.

Hierarchy Level: Various

Format:

\$ THIS IS A COMMENT

Solution Control Command: **ANALYZE**

Description: The first command in the ANALYZE subpacket

Hierarchy Level: Type of run

Format:

ANALYZE

Solution Control Command: **BLAST**

Description: Invokes the blast discipline

Hierarchy Level: Discipline

Format and Examples:

BLAST type (BLCOND = i, TSTEP = j)

BLAST MODAL (BLCOND = 20, TSTEP = 30)

<u>Option</u>	<u>Meaning</u>
type	DIRECT or MODAL. Selects the solution approach. Default = DIRECT .
i	Set identification of a BLAST bulk data entry used to specify nuclear blast parameters.
j	Set identification of TSTEP bulk data entries used to specify time step data for the blast response analysis.

Remarks:

1. **BLAST** must not appear in the **OPTIMIZE** subpacket.

Solution Control Command: **BOUNDARY**

Description: Specifies the displacement sets and related data used in a particular boundary condition.

Hierarchy Level: Boundary condition

Format and Examples:

**BOUNDARY MPC = i, SPC = j, REDUCE = k, SUPPORT = l, METHOD = m,
DYNRED = n, INERTIA = o, ESET = p, K2GG = q, M2GG = r**

BOUNDARY SPC = 6

BOUNDARY SPC = 10, REDUCE = 20, SUPPORT = 30

BOUNDARY SPC = 12, DYNRED = 100, INERTIA = 100, K2GG = FUSSTIFF

Option	Meaning
i	Set identification of a multipoint constraint set. Invokes MPC and MPCADD bulk data entries. (Integer>0)
j	Set identification of a single point constraint set. Invokes SPC, SPC1 and SPCADD bulk data entries. (Integer>0)
k	Set identification of a static condensation set. Invokes ASET, ASET1, OMIT and OMIT1 bulk data entries. (Integer>0)
l	Set identification of the free body support. Invokes SUPORT bulk data entries. (Integer>0)
m	Set identification of the RIGR bulk data entry to be used. (Integer>0)
n	Selects the dynamic reduction parameters from the DYNRED bulk data entry (Integer>0)
o	Selects the JSET1 bulk data entries identifying inertia relief degrees of freedom for performing dynamic reduction (Integer>0)
p	Set identification of the extra degrees of freedom for the boundary condition. Invokes EPOINT bulk data entries. (Integer>0)
q	Selects the direct input stiffness matrix in the g-set. This matrix will be added to KGG for this boundary condition. Refers to a DMI or DMIG Bulk Data entry.
r	Selects the direct input mass matrix in the g-set. This matrix will be added to MGG for this boundary condition. Refers to a DMI or DMIG Bulk Data entry.

Remarks:

1. Note that the REDUCE and ESET set specifications are innovative relative to NASTRAN.
2. The bulk data entries will not be used in ASTROS unless selected in Solution Control.
3. None of the options are required but at least one must appear.
4. K2GG and M2GG affect the system stiffness and mass matrices, respectively, for all disciplines within the boundary condition.
5. K2GG and M2GG names will typically refer to DMI or DMIG entries but may refer to any data base matrix entity of the proper dimension.

Solution Control Command: **END**

Description: Indicates the end of a subpacket.

Hierarchy Level: End

Format:

END

Remarks:

1. The **ANALYZE** and **OPTIMIZE** subpackets each require an **END** command.

Solution Control Command: **FLUTTER**

Description: Invokes the flutter analysis discipline

Hierarchy Level: Discipline

Format and Examples:

**FLUTTER (FLCOND = i, DCONSTRAINT = j, CONTROL = k, K2PP = l, M2PP = m,
B2PP = n , TFL = o, DAMPING = p)**

FLUTTER (FLCOND = 100)

FLUTTER (FLCOND=100,CONTROL=AILERON,K2PP=KAIL,M2PP=MAIL,TFL=5)

<u>Option</u>	<u>Meaning</u>
i	Set identification of a FLUTTER bulk data entry that provides flutter parameters.
j	Set identification of a DCONFLT bulk data entry that defines flutter constraint conditions.
k	Selects the input matrix for splining the extra points to the aerodynamic model. Refers to a DMI bulk data entry.
l	Selects the direct input stiffness matrix. Refers to a DMI or DMIG bulk data entry.
m	Selects the direct input mass matrix. Refers to a DMI or DMIG bulk data entry.
n	Selects the direct input damping matrix. Refers to a DMI or DMIG bulk data entry.
o	Selects the transfer function set to be added to the input matrices. Refers to TF bulk data entries.
p	Set identification of VSDAMP and/or TABDMP bulk data entries that define damping data.

Remarks:

1. The **FLCOND** option is required, all others are optional.
2. **M2PP**, **B2PP** and **K2PP** and **CONTROL** names will typically refer to **DMI** and **DMIG** entries, but may refer to any existing database entity of the proper dimension.
3. The use of the **CONTROL** matrix requires that extra points be defined in the boundary condition.

Solution Control Command: **FREQUENCY**

Description: Invokes the frequency response analysis discipline

Hierarchy Level: Discipline

Format and Examples:

FREQUENCY type (**DLOAD** = i, **FSTEP** = j, **GUST** = k, **K2PP** = l, **M2PP** = m,
 B2PP = n, **TFL** = o, **DAMPING** = p)

FREQUENCY DIRECT (**DLOAD** = 10, **FSTEP** = 20)

FREQUENCY MODAL (**DLOAD**=100,**FSTEP**=30,**K2PP**=**KFREQ**,**M2PP**=**MFREQ**,**TFL**=5)

FREQUENCY DIRECT (**DLOAD** = 100, **FSTEP** = 20 , **GUST** = 55)

Option	Meaning
type	Selects the solution approach from DIRECT or MODAL .
i	Set identification of a DLOAD bulk data entry.
j	Set identification of frequency bulk data entries (FREQ , FREQ1 , or FREQ2) that define the frequency steps for the analysis.
k	Set identification of a GUST bulk data entry which defines the gust parameters.
l	Selects the direct input stiffness matrix. Refers to a DMI or DMIG bulk data entry.
m	Selects the direct input mass matrix. Refers to a DMI or DMIG bulk data entry.
n	Selects the direct input damping matrix. Refers to a DMI or DMIG bulk data entry.
o	Selects the transfer function set to be added to the input matrices. Refers to TF bulk data entries.
p	Set identification of VSDAMP and/or TABDMP bulk data entries that define damping data.

Remarks:

1. The **FREQUENCY** discipline does not generate design constraints for optimization.
2. **type**, **DLOAD** and **FSTEP** are required.
3. No more than one **FREQUENCY** analysis can be done in a single boundary condition.
4. **M2PP**, **B2PP** and **K2PP** names will typically refer to **DMI** and **DMIG** entries, but may refer to any existing database entity of the proper dimension.

Solution Control Command: **LABEL**

Description: Provides identifying information on subcase output.

Hierarchy Level: Label information

Format and Example:

LABEL = n

LABEL = SYMMETRIC MANEUVER LOAD

<u>Option</u>	<u>Meaning</u>
n	Any descriptive message that the user wishes to use to distinguish output.

Remarks:

1. **LABEL** information is used until it is superseded.
2. The **LABEL** command is optional.
3. Labels are limited to no more than 72 characters.

Solution Control Command: **MODES**

Description: Selects the Normal Modes discipline.

Hierarchy Level: Discipline

Format and Examples:

MODES (DCONS = n)

MODES

MODES (DCONS = 10)

<u>Option</u>	<u>Meaning</u>
n	Set identification of DCONFRQ bulk data entries which define frequency constraints for the optimization task.

Remarks:

1. Only one modal analysis can be performed in a boundary condition using the **BIGR** bulk data entry selected on the **BOUNDARY** command.

Solution Control Command: **NPSAERO**

Description: Invokes the nonplanar static aerodynamicis discipline

Hierarchy Level: Discipline

Format and Examples:

NPSAERO [symtype] (TRIM = m)

NPSAERO (TRIM = 60)

NPSAERO ANTISYMMETRIC (TRIM = 70)

<u>Option</u>	<u>Meaning</u>
symtype	Selects the symmetry type for the subcase from SYMMETRIC or ANTISYMMETRIC . (Default is SYMMETRIC)
m	Set identification of a TRIM bulk data entry which provides flight condition information.

Remarks:

1. **TRIM** is required.
2. **NPSAERO** analyses may be freely combined with all other **ASTROS** disciplines.

WARNING
This feature is not working correctly

Solution Control Command: **OPTIMIZE**

Description: Invokes the ASTROS design capability

Hierarchy Level: Type of boundary condition

Format and Examples:

```
OPTIMIZE STRATEGY = ((m1,niter1),(m2,niter2),(m3,niter3)), MAXITER = n,  
                    MOVLIM = o, WINDOW = p, OCMOVLIM = q, ALPHA = r,  
                    CNVRGLIM = s, NRFAC = t, EPS = u
```

OPTIMIZE

```
OPTIMIZE MAXITER = 10, NRFAC = 0.6, EPS = -.05, MOVLIM = 1.3
```

```
OPTIMIZE STRATEGY = FSD, ALPHA = 0.8, MAXFSD = 10
```

```
OPTIMIZE STRATEGY = (FSD,3), ALPHA = 0.8, MAXFSD = 10
```

```
OPTIMIZE STRATEGY = ((FSD,3),OC), ALPHA = 0.8, MAXFSD = 10
```

Option

Meaning

m1,m2,m3	The strategy to be used in optimization. One of MP for math programming methods, OC for generalized optimality criteria or FSD for fully stressed design. The order of input on the strategy command is the order that will be used. Each strategy MP , OC or FSD may only appear once. Default for m1= MP . (Only MP methods will be used)
niter1 niter2, niter3	The number of iterations for m1, m2 and m3 respectively. The default for each is to use the last named method for those iterations remaining up to MAXITER . If MAXITER is less than the sum of specified iterations, ASTROS will warn the user but stop at MAXITER iterations.

STRATEGY	MP for iterations 1 thru MAXITER
STRATEGY = MP	MP for iterations 1 thru MAXITER
STRATEGY = FSD, 5	FSD for iterations 1 thru 5 MP for iterations 6 thru MAXITER
STRATEGY = ((FSD, 5), OC)	FSD for iterations 1 thru 5 OC for iterations 6 thru MAXITER
STRATEGY = ((FSD, 5), (OC, 5))	FSD for iterations 1 thru 5 OC for iterations 6 thru 10 MP for iterations 11 thru MAXITER

n	The maximum number of iterations to be performed using MP , FSD or OC . Default = 15.
o	The move limit applied to local design variables in MP . The local variable after each redesign will lie between t/MOVLIM and $t \cdot \text{MOVLIM}$ where t is the initial value. Default = 2.0, must be greater than 1.0.

p The window around zero in which the **MOVLIM** bound is overridden to allow the local variable to change sign. If **WINDOW=0.0**, the local variable may not change sign. If **WINDOW** is nonzero, the half width of a band around zero, **EPS** is computed

$$EPS = WINDOW/100 * MAX (ABS(TMIN) , ABS(TMAX))$$

If the local variable falls within the band, the new minimum or maximum for the current iteration is changed to lie on the other side of zero from the local variable. The bandwidth **EPS** is a percentage of the larger of **TMAX** or **TMIN** where **WINDOW** specifies the percentage. Default = 0.0, must be greater than or equal to 0.0.

q The move limit applied to local design variables in **OC**. The local variable after each redesign will lie between $t/MOVLIM$ and $t*MOVLIM$ where t is the initial value. Default = 10000.0, must be greater than 1.0. This default implies no move limits.

r Exponential move limit for **FSD**. Numbers less than 1.0 result in a smaller move with smoother convergence. Ignored if **STRAT=MP**, Default = 0.90, must be greater than 0.0

s Convergence limit specifying the maximum percentage change in the objective function that can be considered converged. Default = 0.50, must be greater than 0.0.

t Constraint retention factor for **MP** and **OC** methods. The number of active constraints will be at least **NRFAC** times the number of design variables. Default = 3.0.

u Constraint retention parameter in which all constraints having a value greater than **EPS** will be considered active. Default = -0.10

Remarks:

1. None of the options are required
2. **MAXITER** and **CNVRGLIM** are global parameters that apply to the **MP**, **FSD** and **OC** strategies.
3. **MOVLIM** and **WINDOW** control the move limits for **MP**. **WINDOW** is only useful for **LOCAL** design variables that need to cross between positive and negative values.
4. **NRFAC** and **EPS** control the constraint deletion algorithm for **MP** and **OC**, both values are always applied.

Solution Control Command: **PRINT**

Description: Specifies the required output file processing for the print file

Hierarchy Level: Various

Format and Examples:

```
PRINT      (Form,FREQ=a,ITER=b,MODE=c,TIME=d)
           ACCE (Form,FREQ=a,ITER=b,TIME=d) = e,
           AIRD (Form,ITER=b,MODE=c,TIME=d) = f,
           CGRA (ITER=b) = h,
           DCON (ITER=b) = i,
           DISP (Form,FREQ=a,ITER=b,MODE=c,TIME=d) = j,
           ENER (Form,FREQ=a,ITER=b,MODE=c,TIME=d) = k,
           FORC (Form,FREQ=a,ITER=b,MODE=c,TIME=d) = l,
           GDES (ITER=b) = m,
           GPFO (Form,FREQ=a,ITER=b,MODE=c,TIME=d) = n,
           GPWG (ITER=b) = n1,
           KSNS (ITER=b) = o,
           LDES (ITER=b) = p,
           LOAD (Form,FREQ=a,ITER=b,MODE=c,TIME=d) = q,
           MASS (ITER=b) = r,
           MSNS (ITER=b) = t,
           OGRA (ITER=b) = u,
           QHH  (ITER=b,MODE=c) = x,
           QHJ  (ITER=b,MODE=c) = y,
           ROOT (Form,ITER=b,MODE=c) = z,
           SPCF (Form,FREQ=a,ITER=b,MODE=c,TIME=d) = aa,
           STIF (ITER=b) = ab,
           STRA (Form,FREQ=a,ITER=b,MODE=c,TIME=d) = ac,
           STRE (Form,FREQ=a,ITER=b,MODE=c,TIME=d) = ad,
           TPRE (ITER=b) = ae,
           VELO (Form,FREQ=a,ITER=b,MODE=c,TIME=d) = af
           TRIM

PRINT DISP = ALL

PRINT (RECT,MODE=10,ITER=20) DISP(ITER=LAST) = 6, ENERGY(POLA) = 10

PRINT (MODE=NONE)
```

Options	Meaning
Form	RECT or POLA requests output in RECTangular or POLar format (See Remarks 1 and 2).
a	Set identification of a FREQLIST bulk data entry that is used to request the frequencies at which output is to be printed (See Remark 2).
b	Set identification of an ITERLIST bulk data entry that is used to request the optimization iterations at which output is to be printed (See Remark 2).
c	Set identification of a MODELIST bulk data entry that is used to request the modes at which output is to be printed (See Remark 2).

d	Set identification of a TIMELIST bulk data entry that is used to request the times at which output is to be printed (See Remark 2).
e	Set identification of a GRIDLIST bulk data entry that is used to request the grid points at which accelerations are to be printed.
f	Set identification of an ELEMLIST bulk data entry that is used to request the aerodynamic box elements at which displacements for the aerodynamic model are to be printed.
h	Set identification of an DCONLIST bulk data entry that is used to request the the subset of active constraints for which gradients are to be printed (See Remark 2).
i	Set identification of an DCONLIST bulk data entry that is used to request the the subset of active constraints which are to be printed (See Remark 2).
j	Set identification of a GRIDLIST bulk data entry that is used to request the grid points at which displacements are to be printed.
k	Set identification of an ELEMLIST bulk data entry that is used to request the elements for which strain energies are to be printed.
l	Set identification of an ELEMLIST bulk data entry that is used to request the elements for which forces are to be printed.
m	Set identification of a GDVLIST bulk data entry that is used to request the global design variable IDs for which global design variables are to be printed.
n	Set identification of a GRIDLIST bulk data entry that is used to request the grid points at which grid point forces are to be printed.
n1	Either ALL or NONE depending on whether the GPWG is to be computed/printed. If a GPWG entry is in the Bulk Data file, it will be used by the algorithm.
o	Set identification of an LDVLIST and/or a GDVLIST bulk data entry that is used to request the design variables for which stiffness sensitivities are to be printed.
p	Set identification of an LDVLIST bulk data entry that is used to request the local design variable IDs for which local design variables are to be printed.
q	Set identification of a GRIDLIST bulk data entry that is used to request the grid points at which applied loads are to be printed.
r	Set identification of a GRIDLIST bulk data entry that is used to request the grid points degrees of freedom for which the mass matrix is to be printed.
t	Set identification of an LDVLIST and/or a GDVLIST bulk data entry that is used to request the design variables for which mass sensitivities are to be printed.
u	Set identification of a GDVLIST bulk data entry that is used to request the design variables for which objective function gradients are to be printed.
x	Set identification of an ELEMLIST bulk data entry that is used to request the aerodynamic elements for which QHH is to be printed.
y	Set identification of an ELEMLIST bulk data entry that is used to request the aerodynamic elements for which QHJ is to be printed.
z	Set identification of an MODELIST bulk data entry that is used to request the modes for which flutter and normal modes eigenvalue results are to be printed.
aa	Set identification of a GRIDLIST bulk data entry that is used to request the grid points at which SPC forces are to be printed.
ab	Set identification of a GRIDLIST bulk data entry that is used to request the grid points degrees of freedom for which the stiffness matrix is to be printed.

ac	Set identification of an ELEMLIST bulk data entry that is used to request the elements at which strains are to be printed.
ad	Set identification of an ELEMLIST bulk data entry that is used to request the elements for which stresses are to be printed.
ae	Set identification of an ELEMLIST bulk data entry that is used to request the aerodynamic elements for which the pressure coefficients at aeroelastic trim are to be printed.
af	Set identification of a GRIDLIST bulk data entry that is used to request the grid points at which velocities are to be printed.

Remarks:

1. **Form** is an optional parameter for printing complex data. **RECT**angular data outputs complex data with real and imaginary components while **POLAR** outputs complex data using magnitude and phase.
2. If used with the **PRINT** command, all data that are not otherwise specified use the requested **Form**, **FREQ**, **ITER**, **MODE**, and **TIME**, if applicable for that type of data. If used with an option, **Form**, **FREQ**, **ITER**, **MODE**, and **TIME** override the global request. Options **a** through **af** can be either **ALL**, **NONE**, or a positive integer, and additionally, option **b** (**ITER**) can be **LAST**, and options **h** (**CGRA**) and **i** (**DCON**) can be **ACTIVE**. **ALL** requests all values. **NONE** turns off a request from a previous hierarchy while an integer value refers to a bulk data entry. **LAST** requests that output be printed for only the final value in a list. For example, **ITER=LAST** selects output for the final iteration in an optimization. **ACTIVE** selects the active constraints.
3. **HIST** and **TRIM** are toggles. If they are present, the specified data are printed. **TRIM** indicates that stability derivative data associated with an aeroelastic trim are to be printed. **HIST** indicates that the design iteration history summary is to be printed.
4. Aerodynamic macro elements are selected indirectly. A macro element is chosen by selecting one or more aerodynamic box elements contained within the macro element.

Solution Control Command: PUNCH

Description: Specifies the required output file processing for the punch file

Hierarchy Level: Various

Format and Examples:

PUNCH (Form, FREQ=a, ITER=b, MODE=c, TIME=d)
ACCE (Form, FREQ=a, ITER=b, TIME=d) = e,
AIRD (Form, ITER=b, MODE=c, TIME=d) = f,
CGRA (ITER=b) = h,
DCON (ITER=b) = i,
DISP (Form, FREQ=a, ITER=b, MODE=c, TIME=d) = j,
ENER (Form, FREQ=a, ITER=b, MODE=c, TIME=d) = k,
FORC (Form, FREQ=a, ITER=b, MODE=c, TIME=d) = l,
GDES (ITER=b) = m,
GPFO (Form, FREQ=a, ITER=b, MODE=c, TIME=d) = n,
GPWG (ITER=b) = nl,
KSNS (ITER=b) = o,
LDES (ITER=b) = p,
LOAD (Form, FREQ=a, ITER=b, MODE=c, TIME=d) = q,
MASS (ITER=b) = r,
MODEL (ITER=ag) = ah
MSNS (ITER=b) = t,
OGRA (ITER=b) = u,
QHH (ITER=b, MODE=c) = x,
QHJ (ITER=b, MODE=c) = y,
ROOT (Form, ITER=b, MODE=c) = z,
SPCF (Form, FREQ=a, ITER=b, MODE=c, TIME=d) = aa,
STIF (ITER=b) = ab,
STRA (Form, FREQ=a, ITER=b, MODE=c, TIME=d) = ac,
STRE (Form, FREQ=a, ITER=b, MODE=c, TIME=d) = ad,
TPRE (ITER=b) = ae,
VELO (Form, FREQ=a, ITER=b, MODE=c, TIME=d) = af
TRIM

PUNCH DISP = ALL

PUNCH (RECT, MODE=10, ITER=20) DISP(ITER=LAST) = 6, ENERGY(POLA) = 10

PUNCH (MODE=NONE)

Options	Meaning
Form	RECT or POLA requests output in RECTangular or POLAr format (See Remarks 1 and 2).
a	Set identification of a FREQLIST bulk data entry that is used to request the frequencies at which output is to be punched (See Remark 2).
b	Set identification of an ITERLIST bulk data entry that is used to request the optimization iterations at which output is to be punched (See Remark 2).
c	Set identification of a MODELIST bulk data entry that is used to request the modes at which output is to be punched (See Remark 2).

- d Set identification of a **TIMELIST** bulk data entry that is used to request the times at which output is to be punched (See Remark 2).
- e Set identification of a **GRIDLIST** bulk data entry that is used to request the grid points at which accelerations are to be punched.
- f Set identification of an **ELEMLIST** bulk data entry that is used to request the aerodynamic box elements at which displacements for the aerodynamic model are to be punched.
- h Set identification of an **DCONLIST** bulk data entry that is used to request the the subset of active constraints for which gradients are to be punched (See Remark 2).
- i Set identification of an **DCONLIST** bulk data entry that is used to request the the subset of active constraints which are to be punched (See Remark 2).
- j Set identification of a **GRIDLIST** bulk data entry that is used to request the grid points at which displacements are to be punched.
- k Set identification of an **ELEMLIST** bulk data entry that is used to request the elements for which strain energies are to be punched.
- l Set identification of an **ELEMLIST** bulk data entry that is used to request the elements for which forces are to be punched.
- m Set identification of a **GDVLIST** bulk data entry that is used to request the global design variable IDs for which global design variables are to be punched.
- n Set identification of a **GRIDLIST** bulk data entry that is used to request the grid points at which grid point forces are to be punched.
- n1 Either **ALL** or **NONE** depending on whether the **GPWG** is to be computed/punched. If a **GPWG** entry is in the Bulk Data file, it will be used by the algorithm.
- o Set identification of an **LDVLIST** and/or a **GDVLIST** bulk data entry that is used to request the design variables for which stiffness sensitivities are to be punched.
- p Set identification of an **LDVLIST** bulk data entry that is used to request the local design variable IDs for which local design variables are to be punched.
- q Set identification of a **GRIDLIST** bulk data entry that is used to request the grid points at which applied loads are to be punched.
- r Set identification of a **GRIDLIST** bulk data entry that is used to request the grid points degrees of freedom for which the mass matrix is to be punched.
- t Set identification of an **LDVLIST** and/or a **GDVLIST** bulk data entry that is used to request the design variables for which mass sensitivities are to be punched.
- u Set identification of a **GDVLIST** bulk data entry that is used to request the design variables for which objective function gradients are to be punched.
- x Set identification of an **ELEMLIST** bulk data entry that is used to request the aerodynamic elements for which **QHH** is to be punched.
- y Set identification of an **ELEMLIST** bulk data entry that is used to request the aerodynamic elements for which **QHJ** is to be punched.
- z Set identification of an **MODELIST** bulk data entry that is used to request the modes for which flutter and normal modes eigenvalue results are to be punched.
- aa Set identification of a **GRIDLIST** bulk data entry that is used to request the grid points at which **SPC** forces are to be punched.
- ab Set identification of a **GRIDLIST** bulk data entry that is used to request the grid points degrees of freedom for which the stiffness matrix is to be punched.

ac	Set identification of an ELEMLIST bulk data entry that is used to request the elements at which strains are to be punched.
ad	Set identification of an ELEMLIST bulk data entry that is used to request the elements for which stresses are to be punched.
ae	Set identification of an ELEMLIST bulk data entry that is used to request the aerodynamic elements for which the pressure coefficients at aeroelastic trim are to be punched.
af	Set identification of a GRIDLIST bulk data entry that is used to request the grid points at which velocities are to be punched.
ag	Specifies the iterations at which the design model will be punched. May be ALL , NONE , LAST , or the set identification of an ITERLIST bulk data entry which specifies the iterations at which to punch the model.
ah	Specifies the portion of the model which will be punched. May be ALL or NONE . (Note: an integer value is accepted and treated as ALL)

Remarks:

1. **Form** is an optional parameter for printing complex data. **RECTangular** data outputs complex data with real and imaginary components while **POLAR** outputs complex data using magnitude and phase.
2. If used with the **PRINT** command, all data that are not otherwise specified use the requested **Form**, **FREQ**, **ITER**, **MODE**, and **TIME**, if applicable for that type of data. If used with an option, **Form**, **FREQ**, **ITER**, **MODE**, and **TIME** override the global request. Options **a** through **af** can be either **ALL**, **NONE**, or a positive integer, and additionally, option **b** (**ITER**) can be **LAST**, and options **h** (**CGRA**) and **i** (**DCON**) can be **ACTIVE**. **ALL** requests all values. **NONE** turns off a request from a previous hierarchy while an integer value refers to a bulk data entry. **LAST** requests that output be printed for only the final value in a list. For example, **ITER=LAST** selects output for the final iteration in an optimization. **ACTIVE** selects the active constraints.
3. **HIST** and **TRIM** are toggles. If they are present, the specified data are punched. **TRIM** indicates that stability derivative data associated with an aeroelastic trim are to be punched. **HIST** indicates that the design iteration history summary is to be punched.
4. Aerodynamic macro elements are selected indirectly. A macro element is chosen by selecting one or more aerodynamic box elements contained within the macro element.

Solution Control Command: **SAERO**

Description: Invokes the static aerodynamics discipline

Hierarchy Level: Discipline

Format and Examples:

SAERO [symtype] (TRIM = k , DCON = 0 , STRESS = m , STRAIN = n)

SAERO (TRIM = 60)

SAERO ANTISYMMETRIC (TRIM = 70 , STRESS = 100)

Option	Meaning
symtype	Selects the symmetry type for the subcase from SYMMETRIC or ANTISYMMETRIC . (Default is SYMMETRIC)
k	Set identification of a TRIM bulk data entry which provides flight condition information.
m	Set identification for stress constraints as defined by DCONVM , DCONVMM , DCONVMP , DCONTW , DCONTWM , or DCONTWP bulk data entries.
n	Set identification for strain constraints as defined by DCONEP , DCONEPM , DCONEPP , DCONFT , DCONFTM , or DCONFTP bulk data entries.
o	Set identification for displacement constraints as defined by DCONDSP , DCONTRM , DCONCLA , DCONALE , or DCONSCF bulk data entries.

Remarks:

1. **TRIM** is required. Both **symtyp** and the **CONSTRAINT** section are optional.
2. **SAERO** disciplines may be freely combined with other **ASTROS** disciplines.
3. For compatibility, the alternate form of constraint specification shown below is also allowed. Its use is, however, discouraged.

SAERO [symtype] (TRIM = k), CONSTRAINT (STRESS=m, STRAIN=n, GENERAL=o)

Solution Control Command: **SOLUTION**

Description: The first command in the solution control packet.

Hierarchy Level: Beginning of solution

Format:

SOLUTION

Remarks:

1. One **SOLUTION** command must always appear as the first command of the solution control packet.

Solution Control Command: **STATICS**

Description: Invokes the statics analysis discipline

Hierarchy level: Discipline

Format and Examples:

STATICS (MECH = i, THERMAL = j, GRAVITY = k, DCON = o, STRESS = m, STRAIN = n)

STATICS (MECH = 10)

STATICS (MECH = 4, THERMAL = 6, DCON = 10)

Option	Meaning
i	Set identification for external loads as defined by LOAD , PLOAD , FORCE , FORCE1 , MOMENT , and MOMENT1 bulk data entries.
j	Set identification for temperatures defined by TEMP or TEMPD bulk data entries.
k	Set identification of GRAV bulk data entries which define gravity forces.
m	Set identification for stress constraints defined by DCONVM , DCONVMM , DCONVMP , DCONTW , DCONTWM , or DCONTWP bulk data entries.
n	Set identification for strain constraints defined by DCONEP , DCONEPM , DCONEPP , DCONFT , DCONFTM , or DCONFTP bulk data entries.
o	Set identification of DCONDSP bulk data entries which define displacement constraints.

Remarks.

1. The sum of all the loads forms a single right hand side for a statics analysis.
2. At least one of the load types must be present. The **CONSTRAINT** section is optional.
3. Gravity forces may be included indirectly if referenced by the **LOAD** bulk data entry.
4. For compatibility, the alternate form of constraint specification shown below is also allowed. Its use is, however, discouraged.

STATICS (MECH = i, THERMAL = j, GRAVITY = k),
CONSTRAINT (STRESS=m, STRAIN=n, GENERAL=o)

Solution Control Command: **SUBTITLE**

Description: Defines a subtitle which will appear in the output.

Hierarchy Level: Label information

Format and Example:

SUBTITLE = n

SUBTITLE = SUPERSONIC DESIGN CONDITION

<u>Option</u>	<u>Meaning</u>
n	Any descriptive information can be inserted here

Remarks:

1. **SUBTITLE** information is used until it is superseded.
2. The **SUBTITLE** command is optional.
3. Subtitles are limited to 72 characters.

Solution Control Command: **TITLE**

Description: Defines a title which will appear in the output.

Hierarchy Level: Label information

Format and Examples:

TITLE = n

TITLE = DESIGN OF A FORWARD SWEPT WING MODEL

<u>Option</u>	<u>Meaning</u>
n	Any descriptive information can be inserted here

Remarks:

1. **TITLE** information is used until it is superseded.
2. The **TITLE** command is optional.
3. Titles are limited to no more than 72 characters.

Solution Control Command: TRANSIENT

Description: Invokes the transient analysis discipline

Hierarchy Level: Discipline

Format and Examples:

**TRANSIENT type (DLOAD = i, TSTEP = j, FFT = k, IC = l, GUST = m,
K2PP = n, M2PP = o, B2PP = p, TFL = q, DAMPING = r)**

TRANSIENT MODAL (DLOAD = 10, TSTEP = 20)

TRANSIENT DIRECT (DLOAD=100,TSTEP=30,K2PP=KFREQ,M2PP=MFREQ,IC=45,TFL=5)

TRANSIENT MODAL (DLOAD = 100, TSTEP = 20 , FFT = 999,GUST = 55)

<u>Option</u>	<u>Meaning</u>
type	Selects the solution approach from DIRECT or MODAL .
i	Set identification of a DLOAD bulk data entry.
j	Set identification of TSTEP bulk data entries which provide the time step information for the analysis.
k	Set identification of an FFT bulk data entry which provides parameters to use the Fast Fourier Transform methods in performing the transient analysis.
l	Set identification of IC bulk data entries which define the initial conditions.
m	Set identification of a GUST bulk data entry which defines the gust parameters.
n	Selects the direct input stiffness matrix. Refers to a DMI or DMIG bulk data entry.
o	Selects the direct input mass matrix. Refers to a DMI or DMIG bulk data entry.
p	Selects the direct input damping matrix. Refers to a DMI or DMIG bulk data entry.
q	Selects the transfer function set to be added to the input matrices. Refers to TF bulk data entries.
r	Set identification of VSDAMP and/or TABDMP bulk data entries that define damping data.

Remarks:

1. The **TRANSIENT** discipline does not generate design constraints for optimization.
2. **type**, **DLOAD** and **TSTEP** are required.
3. If **GUST** is present, **FFT** must also be used.
4. Initial conditions, **IC**, are only valid for **DIRECT** analyses. **IC** cannot be used with **GUST** or **FFT**.
5. No more than one **TRANSIENT** analysis can be done in a single boundary condition.
6. **M2PP**, **B2PP** and **K2PP** names will typically refer to **DMI** and **DMIG** entries, but may refer to any existing database entity of the proper dimension.

4. THE BULK DATA PACKET

The bulk data packet provides the ASTROS system with the engineering data needed to perform the specific tasks requested by the user. It contains the model geometries for the structural model, the aerodynamic model(s) and the design model as well as the pool of data from which the solution control requests are made. Finally, specialized information required by the analysis disciplines (e.g., Mach number and reduced frequency pairs for unsteady aerodynamic analyses) is also provided to the system through the bulk data packet. The basic input item is the bulk data entry which is directly analogous to the NASTRAN bulk data *card*. In fact, NASTRAN compatible formats were chosen for the ASTROS bulk data entries whenever possible because modern structures are often analyzed using large NASTRAN finite element models having tens of thousands of lines of bulk data. Further, these large models are usually prepared using software designed specifically to generate NASTRAN models. Thus, by utilizing NASTRAN bulk data structures where possible and by using NASTRAN's bulk data style for the additional engineering data, ASTROS is highly compatible with existing NASTRAN models and with current finite element model generation methods.

Just as in NASTRAN, the bulk data packet begins with the keyword **BEGIN BULK** (which may be abbreviated **BEGIN**) and is terminated by the optional keyword **ENDDATA** or by the end of the input stream. The intervening bulk data entries can appear in any order. An alphabetically sorted listing of the bulk data input will be echoed to the output file unless suppressed by the user through the **BEGIN BULK** command line options.

All the input entries are interpreted by **IFP** through *templates* that are defined as part of the system generation task. The templates provide for basic error checking, establish defaults and direct the placement of the raw data onto the database. The use of templates allows additional entries to be added to the system very simply without software changes. The definition of the templates and the means of adding new entries are documented in the Programmer's Manual. In addition, the complete listing of ASTROS bulk data templates is included in the output summary generated by the SYSGEN system generation utility during the creation of the system database files.

On restart with a bulk data packet in the input stream, the **IFP** module will append the new data onto the data from the previous run(s). There is no provision for deleting existing bulk data except through MAPOL sequence modifications or direct interaction using the ICE program (Reference 7). This restart feature, while limited, can be useful in many instances; e.g. when additional analysis disciplines are desired or when different output requests are desired. The remainder of this section presents the structure of the bulk data entry for ASTROS and discusses some features of the **IFP** module that are useful to the general user. ASTROS bulk data entries have been carefully designed to be NASTRAN compatible, so the NASTRAN User's Manual (Reference 2) has provided much of the information in the following discussion as well as having directed the design of the **IFP** software. The reader is also referred to the ASTROS Programmer's Manual for more information on the **IFP** module and for information on

the addition of new bulk data entries. An additional resource is the MSC/NASTRAN Primer of Reference 3.

4.1. BULK DATA ECHO OPTIONS

There are special options on the **BEGIN BULK** command which allow the user to control the echoing of the Bulk Data. The format of this command is:

BEGIN BULK	<div style="display: inline-block; vertical-align: middle;"> <div style="text-align: center;">ECHO</div> <div style="text-align: center;">NOECHO</div> </div>	<div style="display: inline-block; vertical-align: middle;"> <div style="text-align: center;">PRINT</div> <div style="text-align: center;">PUNCH</div> <div style="text-align: center;">BOTH</div> </div>	<div style="display: inline-block; vertical-align: middle;"> <div style="text-align: center;">SORT</div> <div style="text-align: center;">NOSORT</div> </div>
------------	---	---	---

The following table describes the actions which are performed for the various options.

ECHO	FILE	ORDER	ACTION
ECHO	PRINT	SORT	Sorted echo to the output file.
		UNSORT	Unsorted echo to the output file.
	PUNCH	SORT	Sorted echo to punch file.
		UNSORT	Unsorted echo to punch file.
	BOTH	SORT	Sorted echo to both output and punch files.
		UNSORT	Unsorted echo to both output and punch files.
NOECHO			No echo to either output or punch file.

4.2. FORMAT OF THE BULK DATA ENTRY

Each bulk data entry consists of a required **parent** line followed by a number of optional **continuation** lines. Therefore, a single bulk data entry resides on one or more lines. The basic bulk data line has one mnemonic field of eight characters followed by either eight data fields of eight characters or by four data fields of 16 characters and terminates with an eight character continuation field as shown in Figure 4. The data field size (either eight or 16 characters) is determined by the presence of the optional **large field marker** in the first mnemonic field of each bulk data line. The parent line begins with a character mnemonic identifying the entry followed by 4 or 8 data fields and ending with a continuation field. The continuation lines are identical except that the leading mnemonic field contains a continuation label which is used to link it to its parent line. This structure is identical to that in NASTRAN. One important exception to NASTRAN compatibility is that ASTROS requires that the continuation lines follow continuously from the parent line although the bulk data entries themselves can be in any order. Random placement of continuations in NASTRAN is an artifact from using physical cards that were punched with the bulk data. If the card deck were dropped, the resulting random order still had to be interpretable by the code. This feature no longer needs to be supported in light of modern computer storage methods but NASTRAN compatibility dictated that similar continuation labeling be used.

A continuation line is defined for a bulk data entry that requires more than eight (or four large) data fields. The last field of the parent line is used in conjunction with the first field of the continuation line as an identifier. The parent continuation field can contain any alphanumeric entry while the first field of the continuation line contains a plus (+) as a continuation character in column 1 followed by the last 7 characters from the parent continuation label. For the parent line, the large field marker is an asterisk (*) following the name of the entry which signifies that large data fields are to be used. For continuation lines, the asterisk used as the continuation character plays the role of the large field marker as shown below. Each bulk data line must be either all narrow field or all large field, although separate lines of a single bulk data entry can have different field widths simply by using the proper field marker. This means that the same bulk data entry in wide and narrow formats are functionally identical with no need for separate templates. Unlike NASTRAN, the continuation mnemonics need not be unique among all the bulk data entries in the bulk data packet since there is no provision for randomly sorted continuations.

The input on a bulk data line can either be in fixed format, in which each item must reside within the field to which it belongs, or in free format, in which fields are separated by commas and can be positioned anywhere to the left of the column in which the fixed field would normally start. Free format input is indicated by the appearance of a comma in the first 10 characters of the input line. ASTROS requires that each line (not each bulk data entry) be either all fixed or all free format and that each free format field be separated by a comma. The NASTRAN use of a blank character as a field separator is not

Small Field Entry with a Small Field Continuation

NAME									ABC
+BC									

Small Field Entry with a Large Field Continuation

NAME									ABC
*BC									

Large Field Entry with both Large and Small Field Continuations

NAME*									ABC
*BC									DEF
+EF									

Large Field Entry with a Large Field Continuation

NAME*									ABC
*BC									

Figure 4. Bulk Data Entry Formats

supported. When free format input is used, the continuation lines can reside on the same physical line of input with the continuation labels either included or not as in the following equivalent examples:

```
MKAERO1, 1, , 0.3, 0.5, , , , ABC , +BC, 0.01, 0.05, 0.1, 0.2
```

```
MKAERO1, 1, , 0.3, 0.5, , , , 0.01, 0.05, 0.1, 0.2
```

In the latter case, ASTROS will automatically generate the missing continuation mnemonics. Care must be taken, however, that the first *two* data fields of the continuation line be non-blank. If not, there is an ambiguity as to whether the first continuation field constitutes a continuation label or a data field. This ambiguity causes the IFP to terminate execution with an error indicating that there is a missing continuation line. Free format input in which the parent and continuation lines are broken into separate physical lines or which explicitly include the continuation mnemonics do not suffer this limitation. Free format input is further restricted in that the break between physical lines, if needed, must occur at a break in the logical line, that is, the split must occur between the ending continuation field on the current logical line and the continuation field of the next logical line. This means, for the preceding example, that the first example entry could be broken into two lines between the ABC and +BC fields but nowhere else. When an entry is broken into multiple physical lines, the continuation mnemonics must be supplied. Obviously, fixed format input requires continuation mnemonics for any bulk data entries having continuation lines.

4.3. DATA FIELD FORMATS

The interior fields of a bulk data line can contain either integer data, real data, character data or certain combinations (e.g. either integer or real data). The template for each entry defines which types of data are acceptable in each field. Each data item is limited to the number of characters that fit in the length of the field. For narrow width fields no more than eight characters can be used in the data item. Unlike NASTRAN, any extra characters will spill to the next field and will result in IFP errors, there is no provision for rounding real data to fit the field size.

In order to be considered valid, the data item must first satisfy the data type requirement as specified on the template. Real numbers, including zero, must contain a decimal point, although there are a number of formats supported. For example, the real number 3.1 may be encoded as shown or as 3.1E0, +3.1D00, 0.31E1, or 3.1+0. Unlike NASTRAN, however, there *cannot* be embedded blanks anywhere in the real number and a D edit descriptor is treated as a single precision number until actually loaded to a double precision relational attribute. Blank fields that do not have other defaults specified on the template, will be interpreted as blank characters, an integer zero or a real zero as required. Integer values must be formed from the ten decimal digits with an optional leading plus or minus sign. Character data consist of any combination of alphanumeric characters including any digits, decimal points, etc., with no restriction that the first character be alphabetic.

4.4. ERROR CHECKING IN THE INPUT FILE PROCESSOR

As mentioned in the preceding subsection, the IFP module performs basic error checking to ensure that the input data is of the correct type. In addition, the templates provide for error checks that enable the IFP to check that the data satisfy particular requirements. For example, the IFP can be

directed to require that a particular value be greater than zero or be one of a finite number of selections. At its most complex, the bulk data processor checks to ensure specific relationships among data on a single bulk data entry. It is important to understand, however, that no error checks occur in the **IFP** to ensure that references to, and interrelationships among, multiple bulk data entries are satisfied. These more complex checks occur in subsequent engineering modules. A complete description of the available template error checks and the mechanism provided to add additional error checks is presented in the Programmer's Manual. The reader may find it helpful to study this documentation since the bulk data packet and the bulk data entries are closely linked to the software in both the **SYSGEN** utility and the **IFP** module.

4.5. BULK DATA ENTRY SUMMARY

This section contains a summary of all the bulk data entries in the **ASTROS** system separated into logically related groups. The groups are composed of either model definition entries, subcase definition entries or general list entries. This is followed by a detailed description of each of the entries listed in this section. Section 4.6 discusses the differences between **NASTRAN** and **ASTROS** for those entries that have been changed or are completely different than in **NASTRAN** but that use the same mnemonic and serve a similar purpose. Entries indicated by * are unchanged from **NASTRAN**.

4.5.1. Aerodynamic Load Transfer

ATTACH	Rigid load transfer definition.
SET1 *	A structural grid point list for spline interpolation or a mode list for omitting normal modes in flutter analysis.
SET2 *	Structural grid point list in term of aerodynamic macroelements.
SPLINE1 *	Surface spline definition for out-of-plane motion.
SPLINE2 *	Beam spline definition for interpolating panels and bodies.

4.5.2. Applied Dynamic Loads

DLAGS	Time and phase lag definition for a spatial load.
DLOAD *	Linear combination of dynamic load sets.
DLONLY	Direct definition of dynamic spatial load.
GUST	Stationary vertical gust definition.
RLOAD1	Frequency dependent dynamic load definition.
RLOAD2	Frequency dependent dynamic load definition.
TABLED1	Tabular function definition for dynamic load generation.
TLOAD1	Time dependent dynamic load definition.
TLOAD2	Time dependent dynamic load definition.

4.5.3. Applied Static Loads

FORCE *	Definition of a concentrated load at a grid point.
FORCE1 *	Definition of a concentrated load at a grid point.
GRAV *	Definition of an acceleration vector for gravity loads.

LOAD*	Definition of linear load combinations.
MOMENT*	Definition of a moment at a grid point.
MOMENT1*	Definition of a moment at a grid point.
PLOAD*	Definition of a pressure load over an area.
TEMP*	Definition of a temperature at a structural node.
TEMPD*	Definition of default nodal temperatures.

4.5.4. Boundary Condition Constraints

ASET*	Analysis set definition.
ASET1*	Analysis set definition.
DYNRED	Dynamic reduction parameters.
JSET	Inertia relief mode shape parameter definition.
JSET1	Inertia relief mode shape parameter definition.
MPC*	Multipoint constraint definition.
MPCADD*	Definition of combinations of MPC sets.
OMIT	Omit set definition.
OMIT1	Omit set definition.
RBAR	Rigid bar element
RBE1	Rigid body element
RBE2	Rigid body element
RBE3	Rigid body element
RROD	Rigid rod element
SPC*	Single point constraint/enforced displacement definition.
SPC1*	Single point constraint definition.
SPCADD*	Definition of combinations of SPC sets.
SUPPORT	Definition of coordinates for determinate reactions.

4.5.5. Design Constraints

DCONALE	Aileron effectiveness constraint definition.
DCONCLA	Lift effectiveness constraint definition.
DCONDSP	Displacement constraint definition.
DCONEP	Principal strain constraint definition.
DCONEPM	Principal strain constraint definition.
DCONEPP	Principal strain constraint definition.
DCONFLT	Flutter constraint definition.
DCONFRQ	Modal frequency constraint definition.
DCONFT	Fiber/transverse strain constraint definition.
DCONFTM	Fiber/transverse strain constraint definition.
DCONFTP	Fiber/transverse strain constraint definition.

DCONSCF	Flexible stability coefficient constraint definition.
DCONTHK	Thickness constraint definition for use with shape function design variable linking.
DCONTRM	Aeroelastic trim parameter constraint definition.
DCONTW	Tsai-Wu stress constraint definition.
DCONTWM	Tsai-Wu stress constraint definition.
DCONTWP	Tsai-Wu stress constraint definition.
DCONVM	Von-Mises stress constraint definition.
DCONVMM	Von-Mises stress constraint definition.
DCONVMP	Von-Mises stress constraint definition.

4.5.6. Design Variables, Linking and Optimization Parameters

DESELM	Unique physical design variable definition.
DESVARP	Linked physical design variable definition
DESVARs	Linked shape function design variable definition.
ELIST	Element list for physical linking.
MPPARM	Mathematical programming default parameter override.
OCPARAM	Optimality criteria default parameter override.
PLIST	Physical design variable linking definition.
SHAPE	Definition of element linking factors to define a shape variable.

4.5.7. Geometry

CORD1C*	Cylindrical coordinate system definition.
CORD1R*	Rectangular coordinate system definition.
CORD1S*	Spherical coordinate system definition.
CORD2C*	Cylindrical coordinate system definition.
CORD2R*	Rectangular coordinate system definition.
CORD2S*	Spherical coordinate system definition.
EPOINT*	Extra point definition for dynamics.
GRDSET*	Default parameters for fields on the GRID entry.
GRID*	Grid point location and coordinate system selection.
SPOINT*	Scalar point definition.

4.5.8. Material Properties

MAT1*	Isotropic elastic properties definition.
MAT2*	Two-dimensional anisotropic properties definition.
MAT8*	Orthotropic properties definition.
MAT9*	Anisotropic properties definition for isoparametric hexahedral elements.

4.5.9. Miscellaneous Inputs

\$*	Commentary data.
CONVERT	Conversion factor definitions.
DMI	Direct matrix input.
DMIG	Direct matrix input at structural nodes.
MFORM	Mass matrix form (LUMPED or COUPLED).
SAVE	List of database entities not to be purged.
SEQGP*	Structural set resequencing definition.

4.5.10. Output Selection Lists

DCONLIST	A list of design constraints for which constraint value and/or gradient output are desired.
ELEMLIST	List of elements for element dependent output.
FREQLIST	List of frequency steps for which output is desired.
GDVLIST	List of global design variables for which output is desired.
GPWG	Definition of the location to perform grid point weight generation
GRIDLIST	List of nodes for nodal dependent output.
ITERLIST	List of iteration steps for which output is desired
LDVLIST	List of local design variables for which output is desired.
MODELIST	List of normal modes for which output is desired.
TIMELIST	List of time steps for which output is desired.

4.5.11. Steady Aerodynamics

AEROS	Reference parameters
AEFACT	List of real parameters.
AESURF	Aerodynamic control surface definition.
AIRFOIL	Airfoil property definition.
AXSTA	Body axial station parameter definition.
BODY	Body configuration definition.
CAERO6	Macroelement (panel) definition.
CONEFFS	Definition of static aerodynamic control effectiveness
CONLINK	Definition of linked control surfaces.
PAERO6	Body parameter definition.

4.5.12. Structural Element Connection

BAROR	Definition of default parameters for the CBAR bar element.
CBAR	Prismatic beam element.
CELAS1	Scalar elastic spring element.
CELAS2	Scalar elastic spring element.

CIHEX1*	Linear isoparametric hexahedral element.
CIHEX2*	Quadratic isoparametric hexahedral element.
CIHEX3*	Cubic isoparametric hexahedral element.
CMASS1	Scalar mass element.
CMASS2	Scalar mass element.
CONM1*	Direct 6 x 6 mass matrix definition at a structural node.
CONM2	Concentrated mass at a structural node.
CONROD	Rod element.
CQDMEM1	Isoparametric quadrilateral membrane element.
CQUAD4	Isoparametric quadrilateral element with bending and membrane stiffness.
CROD	Rod element.
CSHEAR	Shear panel.
CTRIA3	Isoparametric triangular element with bending and membrane stiffness.
CTRMEM	Constant strain triangular membrane element.

4.5.13. Structural Element Properties

PBAR	Prismatic beam element.
PCOMP	Composite laminate definition for CQDMEM1 , CQUAD4 , CTRIA3 , and CTRMEM elements.
PCOMP1	Composite laminate definition for CQUAD4 and CTRIA3 elements.
PCOMP2	Composite laminate definition for CQUAD4 and CTRIA3 elements.
PELAS	Scalar elastic spring element.
PIHEX*	Linear, quadratic and cubic isoparametric hexahedral element.
PMASS	Scalar mass element
PQDMEM1	Isoparametric quadrilateral membrane element.
PROD	Rod element.
PSHEAR	Shear panel.
PSHELL	Definition of shell element properties for CQUAD4 and CTRIA3 elements.
PTRMEM	Constant strain triangular membrane element.

4.5.14. Unsteady Aerodynamics

AERO	Reference parameters
CAERO1*	Aerodynamic macroelement (panel) definition.
CAERO2*	Body configuration definition.
CONEFF	Definition of flutter aerodynamic control effectiveness
FLFACT*	Parameter definition for flutter analysis.
MKAERO1	Table of symmetries, Mach numbers, and reduced frequencies.
MKAERO2	Table of symmetries, Mach numbers, and reduced frequencies.

PAERO1 *	Association between bodies and macroelements.
PAERO2 *	Body cross-section property definition.

4.5.15. Discipline Dependent Problem Control

The following bulk data entries are the controlling entries referenced by Solution Control in selecting specific disciplines and subcases. In each case, many of these inputs can appear in the bulk data packet with the particular input to be used for the subcase referenced in the Solution Control Packet.

BLAST	Parameters for nuclear blast analyses.
FLUTTER	Basic parameters for flutter analyses.
TRIM	Flight condition for steady aeroelastic trim analyses.
EIGC	Complex eigenvalue extraction parameters
EIGR *	Real eigenvalue extraction parameters.
FFT	Fast Fourier Transform parameter definition.
FREQ *	Frequency step definition for frequency response.
FREQ1 *	Frequency step definition for frequency response.
FREQ2 *	Frequency step definition for frequency response.
IC	Initial condition definition for direct transient response (same as NASTRAN TIC entry).
TABDMP1	Modal damping table for modal dynamic response.
TF *	Dynamic transfer function definition.
TSTEP *	Time step definition for transient response
VSDAMP	Definition of viscous damping based on equivalent structural damping.

4.6. DIFFERENCES BETWEEN ASTROS AND NASTRAN BULK DATA

Some of the bulk data entries listed in the preceding Section do not exist in the MSC/NASTRAN or the COSMIC/NASTRAN versions that guided the definition of the bulk data entries. Some of them do exist in other NASTRAN systems, however; the DYNRED, JSET, JSET1, PCOMP1, and PCOMP2 entries are examples. Others take the place of the NASTRAN PARAM entry which was felt to have been overused to the point where it had lost all utility. Examples of these inputs are the CONVERT, MFORM and VSDAMP entries. The steady aeroelastic model is completely new to ASTROS since NASTRAN uses the same modeling for both steady and unsteady analysis. Also, it was felt that the NASTRAN mechanism for defining dynamic loads was needlessly complicated. Working from the NASTRAN inputs, a simpler, but equally general set of entries was developed. This resulted in the generation of a number of new entries and the modification of others. The definition of the design variables, design variable linking and the design constraints is, of course, completely new for ASTROS.

The majority of the changed entries have been modified to accommodate the design task. In these cases, the bulk data entry is often identical to the NASTRAN version for use in analysis with optional additional fields to specify the design data. The element connectivity and property entries are all examples of this type of change in that additional field(s) have been added to specify the maximum and minimum allowable physical design variable value if shape function design variable linking is used. In

cases where data from NASTRAN preprocessors are used, there are no changes required unless shape function linking is desired.

A more subtle set of changes was required to perform multidisciplinary analysis. In NASTRAN, as was mentioned in the discussion of the Solution Control packet, many parameters were specified as part of the model definition or discipline specification because the code was limited to performing a single analysis of the given discipline. In order to remove these artificial restrictions, these data have been moved to the proper discipline's subcase definition. Examples of this form of modification are the addition of symmetry options to the **MKAERO1**, **GUST**, **FLUTTER**, and **TRIM** entries and the removal of subcase dependent data from the **AERO** entry. Further, the rigid elements, **ASET1**, **OMIT1** and **EPOINT** entries were modified to include a set identification number to enable multiple boundary conditions and multiple control systems to be analyzed simultaneously.

The last type of modification came about because of the nature of the ASTROS database management system. These were limited to the **DMI** and **DMIG** entries for direct loading of database entities. The NASTRAN inputs were not compatible with the ASTROS database and so had to be modified. In fact, these entries, while having the same name as a NASTRAN entry, are completely new entries for ASTROS. A minor additional modification to the input definitions was made for the **TABDMF1** entry to make it more compact and to remove the spurious **ENDT** table termination symbol. In ASTROS, all tabular input entries are terminated when no more data appears and require no specific declaration of the table end.

While a seemingly large number of bulk data entries have been changed relative to their NASTRAN counterparts, in fact only a few have been changed in such a way that the NASTRAN version will not work in analysis. By far, the majority of the modeling bulk data entries are completely unchanged except for certain design variable linking options. In unsteady and steady aerodynamic disciplines care must be taken to account for the subcase dependencies that NASTRAN defined implicitly or with **PARAM** entries. Finally, the use of **ASET** and **OMIT** entries will cause minor problems in that ASTROS requires a set identification for these entries. While this latter restriction can require some effort to fix, the gain in capability simply required that the bulk data entry be modified.

The most serious potential problem using NASTRAN models in ASTROS is that the set of bulk data entries is more limited in ASTROS than in NASTRAN. The ASTROS system has been developed primarily as a multidisciplinary preliminary design tool and does not yet contain the wide range of options supported by a mature code like NASTRAN. The many NASTRAN input entries supporting these options, therefore, have not been defined to the ASTROS system because they are not supported by any ASTROS code. Thus, there will be instances where a NASTRAN input deck will have to be modified to remove these entries which serve no purpose in ASTROS. The majority of these bulk data entries deal with unsupported elements, plotting options, output options, etc., which are not felt to present a major problem. More important is the support for NASTRAN's model definitions, most of which have already been adopted by ASTROS.

4.7. BULK DATA DESCRIPTIONS

This Section contains a complete description of each of the ASTROS Bulk Data entries.

Input Data Entry: \$Comment

Description: For user convenience in inserting commentary material into the unsorted echo of the input Bulk Data Deck. The \$ entry is otherwise ignored by the program. These entries will not appear in a sorted echo.

Format and Example:

1	2	3	4	5	6	7	8	9	10
\$ Followed by any legitimate characters in columns 2-80									
\$ THIS (*,,"\$\$)--/									

Remarks:

1. The comment entry may also be used in the Solution Control packet.

Input Data Entry: AEFACF Aerodynamic Lists

Description: Used to specify lists of real numbers for aeroelastic analysis.

Format and Example:

1	2	3	4	5	6	7	8	9	10
AEFACT	SID	D1	D2	D3	D4	D5	D6	D7	CONT
CONT	D8	D9	-etc-						
AEFACT	97	.3	.7	1.0					

Field

Contents

SID Set identification number (Unique Integer > 0).

Di Number (Real).

Remarks:

1. These factors must be selected by an AIRFOIL, AXSTA, CAERO1 or PAERO1 data entry.
2. Embedded blank fields are forbidden.
3. If used to specify division points, note that there is one more division point than the number of divisions.

Input Data Entry: AERO Aerodynamic Physical Data

Description: Gives basic aerodynamic parameters for unsteady aerodynamic disciplines.

Format and Example:

1	2	3	4	5	6	7	8	9	10
AERO	ACSID	REFC	RHOREF						
AERO	100	300.0	1.1E-7						

Field	Contents
ACSID	Aerodynamic coordinate system identification (Integer ≥ 0 or Blank). See Remark 2.
REFC	Reference length (for reduced frequency) (Real ≥ 0).
RHOREF	Reference density (Real ≥ 0).

Remarks:

1. This entry is required for unsteady aerodynamic disciplines. Only one **AERO** entry is allowed.
2. The **ACSID** must be a rectangular coordinate system. Flow is in the positive x-direction. If blank, the basic coordinate system is used.

Input Data Entry: AEROS Static Aero Physical Data

Description: Gives basic parameters for static aeroelasticity.

Format and Example:

1	2	3	4	5	6	7	8	9	10
AEROS	ACSID	RCSID	REFC	REFB	REFS	GRES	REFD	REFL	

AEROS	10	20	10.	100.	1000.	1			
-------	----	----	-----	------	-------	---	--	--	--

Field	Contents
ACSID	Aerodynamic coordinate system identification (Integer > 0) or blank. See Remark 2.
RCSID	Reference coordinate system identification for rigid body motions (Integer > 0) or blank.
REFC	Reference chord length (Real > 0.0) (D = 1.0)
REFB	Reference span (Real > 0.0) (D = 1.0)
REFS	Reference wing area (Real > 0.0) (D = 1.0)
GRES	Reference grid point for stability derivative calculations (Integer > 0).
REFD	Fuselage reference diameter (Real > 0) or blank (D = 1.0)
REFL	Fuselage reference length (Real > 0) or blank (D = 1.0)

Remarks:

1. This entry is required for static aeroelasticity problems. Only one AEROS entry is allowed.
2. The ACSID must be a rectangular coordinate system. Flow is in the positive x-direction.
3. The RCSID must be a rectangular coordinate system. All degrees of freedom defining trim variables will be defined in this coordinate system.

Input Data Entry AESURF Aerodynamic Control Surface

Description: Specifies an Aerodynamic Control Surface.

Format and Example:

1	2	3	4	5	6	7	8	9	10
AESURF	LABEL	TYPE	ACID	CID	FBOXID	LBOXID			
AESURF	ELEV	SYM	6000		6010	6030			

Field	Contents
LABEL	Unique alphanumeric string of up to eight characters used to identify the control surface
TYPE	Surface type (Character) (Remark 2) SYM symmetric surface ANTISYM antisymmetric surface ASYM Asymmetric surface
ACID	Identification number of the aircraft component (CAERO6) on which the surface lies. (Integer > 0)
CID	Identification number of a rectangular coordinate system whose y-axis defines the hinge line of the control surface. (Integer > 0 or blank)
FBOXID	First aero box on the control surface relative to ACID . (Integer > 0)
LBOXID	Last aero box on the control surface relative to ACID . (Integer > 0)

Remarks:

1. The **LABEL** is arbitrary, but all labels must be unique.
2. The asymmetric surface, **TYPE=ASYM**, is not currently available. Pitch controllers are **TYPE=SYM** while yaw and roll controllers are **TYPE=ANTISYM**.
3. The aerodynamic box numbering scheme is illustrated on the **CAERO1** Bulk Data entry.

Input Data Entry AIRFOIL Airfoil Definition

Description: Defines airfoil properties for the static aerodynamic model.

Format and Example:

1	2	3	4	5	6	7	8	9	10
AIRFOIL	ACID	CMPNT	CP	CHORD	USO/THK	LSO	CAM	RADIUS	CONT
CONT	X1	Y1	Z1	X12	IPANEL				

AIRFOIL	1	WING	1	10	20		30		ABC
+BC	0.0	0.0	0.0	50.0					

Field	Contents
ACID	Associated aircraft component identification number referenced by a matching CAERO6 bulk data entry. (Integer > 0)
CMPNT	Type of aircraft component selected from WING , FIN , and CANARD . (Text) (See Remark 3)
CP	Coordinate system for airfoil. (Integer > 0, or blank) (See Remark 4)
CHORD	Identification number of an AEFACT data entry containing a list of division points (in terms of percent chord) at which airfoil thickness and camber data are specified. (Integer > 0)
USO/THK	Identification number of an AEFACT data entry defining either the upper surface ordinates in percent chord if LSO is not blank, or the half thicknesses about the camber ordinates if CAM is not blank. (Integer > 0, or blank) (See Remark 3)
LSO	Identification number of an AEFACT data entry defining the lower surface ordinates in percent chord. Must be used in conjunction with USO . (Integer > 0, or blank) (See Remark 3)
CAM	Identification number of an AEFACT data entry defining the mean line (camber line) ordinates in percent chord. (Integer) (See Remark 3)
RADIUS	Radius of leading edge in percent chord. (Real ≥ 0.0)
X1, Y1, Z1	Location of the airfoil leading edge in coordinate system CP . (Real, $Y1 \geq 0.0$)
X12	Airfoil chord length in x-axis coordinate of system CP . (Real > 0 or blank)
IPANEL	Identification number of an AEFACT data entry containing a list of chord wise cuts in percent chord for wing paneling. (Integer > 0, or blank)

Remarks:

1. If the **RADIUS** field is blank, a round leading edge of radius zero is used.
2. **IPANEL** is optional and is used when different chord-wise cuts on each end of the panel are desired.

3.

For WING components, the options for USO, LSO, THK and CAM are:	
All Blank	Default flat plat airfoil generated automatically
USO alone	Lower and upper surface ordinates of airfoil are defined with effectively LSO=USO internally generated
USO/LSO	Lower and upper surface ordinates of airfoil are defined; CAM Must be blank
THK/CAM	Half thicknesses about the camber line are defined. LSO Must be blank
USO/LSO/CAM	Illegal over-specification of data
LSO/CAM	Illegal , must use THK field for half thickness
CAM alone	Illegal under-specification of data
For CANARD components, the options are as above except that camber is not allowed so CAM Must be blank	
All Blank	Default flat plat airfoil generated automatically
USO alone	Lower and upper surface ordinates of airfoil are defined with effectively LSO=USO internally generated
USO/LSO	Lower and upper surface ordinates of airfoil are defined; CAM Must be blank
THK/CAM	Illegal specification, CAM must be blank
USO/LSO/CAM	Illegal over-specification of data
LSO/CAM	Illegal , CAM must be blank
CAM alone	Illegal under-specification of data and CAM must be blank for CANARD
For FIN components, the options are very limited: only symmetric airfoils are allowed and they must be entered as an upper surface ordinate (the lower surface ordinates are then defaulted)	
All Blank	Default flat plat airfoil generated automatically
USO alone	Lower and upper surface ordinates of airfoil are defined with effectively LSO=USO internally generated. Only Legal Nonblank Fin Option

4. The basic coordinate system must be used (CP blank). This field exists to allow the addition of user defined coordinate systems in the future.

Input Data Entry: **ASET** Selected Coordinates for the a-set

Description: Defines degrees of freedom that the user desires to place in the analysis set. Used to define the number of independent degrees of freedom.

Format and Examples:

1	2	3	4	5	6	7	8	9	10
ASET	SETID	ID	C	ID	C	ID	C		

ASET	16	2	23	3516					
------	----	---	----	------	--	--	--	--	--

Field	Contents
SETID	The set identification number of the REDUCE set. (Integer > 0)
ID	Grid or scalar point identification number (Integer > 0)
C	Component number, zero or blank for scalar points, any unique combinations of the digits 1 through 6 for grid points.

Remarks:

1. When **ASET** and/or **ASET1** entries are present, all degrees of freedom not otherwise constrained will be placed on the o-set. The o-set is a mutually exclusive set. Degrees of freedom may not be specified on other entries that define mutually exclusive sets.
2. **ASET** entries must be selected in Solution Control (**REDUCE=SETID**) to be used.

Input Data Entry: ASET1 Selected Coordinates for the a-set, Alternate Form

Description: Defines degrees of freedom that the user desires to place in the analysis set. Used to define the number of independent degrees of freedom.

Format and Examples:

1	2	3	4	5	6	7	8	9	10
ASET1	SETID	C	G	G	G	G	G	G	CONT
CONT	G	G	G	-etc-					

ASET1	345	2	1	3	10	9	6	15	ABC
+bc	7	8							

Alternate Form:

1	2	3	4	5	6	7	8	9	10
ASET1	SETI	C	ID1	"THRU"	ID2				

Field	Contents
SETID	The REDUCE set identification number (Integer > 0)
C	Component number (any unique combination of the digits 1 through 6 with no embedded blanks) when point identification numbers are grid points; must be null or zero if point identification numbers are scalar points.
G, ID1, ID2	Grid or scalar point identification numbers (Integer > 0, ID2 > ID1)

Remarks:

1. When **ASET** and/or **ASET1** entries are present, all degrees of freedom not otherwise constrained will be placed in the o-set. The o-set is a mutually exclusive set. Degrees of freedom may not be specified on other entries that define mutually exclusive sets.
2. If the alternate form is used, all points in the sequence ID1 through ID2 are required to exist.
3. **ASET1** entries must be selected in Solution Control (**REDUCE=SETID**) to be used.

Input Data Entry: ATTACH

Description: Defines the aerodynamic control points to be attached to a reference grid for load transfer.

Format and Example:

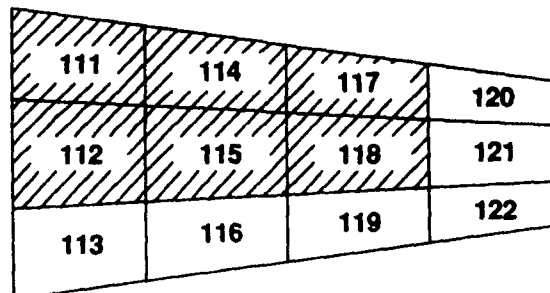
1	2	3	4	5	6	7	8	9	10
ATTACH	EID	MACROID	BOX1	BOX2	RGRID				

ATTACH	100	111	111	118	1				
--------	-----	-----	-----	-----	---	--	--	--	--

Field	Contents
EID	Element identification number (Integer > 0)
MACROID	Element identification of a CAERO1 or PAERO1 element which contains the specified aerodynamic control points (Integer > 0)
BOX1 , BOX2	Starting and final box whose force is to be transferred to the referenced grid (Integer > 0, BOX2 > BOX1)
RGRID	Grid point identification of reference grid point (Integer > 0)

Remarks:

1. The **EID** is used only for error messages.
2. This entry applies to both the steady and unsteady aerodynamic models.
3. The attached aerodynamic boxes are selected as shown below:



Input Data Entry: AXSTA

Description: Defines body axial station parameters. There is one AXSTA for each axial station at which the surface points are defined.

Format and Examples:

1	2	3	4	5	6	7	8	9	10
AXSTA	BCID	XSTA	CBOD	ABOD	LYRAD	LZRAD			
AXSTA	10	10.00	0.5		10	20			

Field	Contents
BCID	Body component identification number (Integer > 0)
XSTA	Value of the x-ordinate of the body station (Real)
CBOD	Value of the z-ordinate of the center line at this station. This defines the body camber (Real).
ABOD	Cross sectional area of the body at this station (Real ≥ 0.0).
LYRAD, LZRAD	Identification number of an AEFACT data entry containing a list of the y-ordinates (z-ordinates) of the body section. (Integer ≥ 0.0)

Remarks:

1. If **ABOD** is present, the body is assumed to be circular and the radial ordinates are computed at **NRAD** (cf. the **BODY** bulk data entry) equal intervals. No **LYRAD** and **LZRAD** data are allowed when **ABOD** is present.
2. If **ABOD** is blank, **LYRAD** and **LZRAD** data must be present.
3. For Pods, **CBOD**, **LYRAD** and **LZRAD** data are not permitted.
4. For the fuselage, **XSTA** is actual x location; for pods, **XSTA** is relative to the **XLOC** value given on the **BODY** bulk data entry.

Input Data Entry: BAROR Simple Beam (BAR) Orientation Default Values

Description: Defines default values for fields 3 and 6 - 8 of the CBAR entry.

Format and Examples:

1	2	3	4	5	6	7	8	9	10
BAROR		PID			X1, GO	X2	X3		
BAROR		39			0.6	2.9	-5.87		

Field	Contents
PID	Identification number of PBAR property entry (Integer > 0 or blank)
Xi	Vector components measured in displacement coordinate system at GA to determine (with the vector from end A to end B) the orientation of the element coordinate system for the bar element (Real or blank)
GO	Grid point identification number (Integer > 0)

Remarks:

1. The contents of fields on this entry will be assumed for any CBAR entry whose corresponding fields are blank.
2. Only one BAROR entry may appear in the user's Bulk Data Deck.

Input Data Entry: BLAST Nuclear Blast Parameters

Description: Defines basic parameters needed for nuclear blast response analysis

Format and Example:

1	2	3	4	5	6	7	8	9	10
BLAST	BLID	ALTITUDE	VELOCITY	WKT	BALT	MACH	DELTAX	DELTAY	CONT
CONT	KGRD	HGRD	SYMXZ	SYMXY	TRSURF	NZ			CONT
CONT	TMIN	TMAX	NTIME	BMIN	BMAX	NBETA			

BLAST	100	10000.	1000.0	1.E5	15000.	0.8			ABC
+BC			1		ELEV	6.0			

Field	Contents
BLID	Set identification number referenced by Solution Control (Integer > 0)
ALTITUDE	Aircraft altitude (Real ≥ 0.0)
VELOCITY	Aircraft velocity (Real ≥ 0.0)
WKT	Weapon yield (Real ≥ 0.0)
BALT	Blast altitude (Real)
MACH	Mach number (Real $0.0 \leq \text{Mach} \leq 1.0$)
DELTAX	X distance from aircraft to blast point (Real)
DELTAY	Y distance from aircraft to blast point (Real)
KGRD	Key denoting presence of the ground (Integer) = 0, no ground = 1 include the ground
HGRD	Height of ground level (Real)
SYMXZ	Symmetry flag for blast analysis about xz plane (Integer)
SYMXY	Symmetry flag for blast analysis about xy plane (Integer)
TRSURF	Label of an AESURF entry used as the trim surface (Character)
NZ	Load factor for the trim calculation (Real)
TMIN	Minimum time used in the definition of polynomial curve fitting (Real > 0.0, or blank, default = 0.10)
TMAX	Maximum time used in the definition of polynomial curve fitting (Real > 0.0, or blank, default = 20.0)
NTIME	Number of time steps (Integer > 0, or blank, default = 20)
BMIN	Minimum Beta value used in the definition of polynomial curve fitting (Real > 0.0, or blank, default = 0.375)

BMAX **Maximum Beta value used in the definition of polynomial curve fitting (Real > 0.0, or blank, default = 10.0)**

NBETA **Number of Beta values (Integer > 0, or blank, default = 7)**

Remarks:

1. In order to be used, the **BLID** must be referenced by Solution Control.
2. The second continuation entry is not required.
3. The default values of **TMIN**, **TMAX**, **NTIME**, **BMIN**, **BMAX** and **NBETA** should be adequate.
4. The symmetry flags refer to the options selected on the **MAKEROI** entries.

Input Data Entry: BODY

Description: Defines body configuration parameters for steady aeroelasticity.

Format and Example:

1	2	3	4	5	6	7	8	9	10
BODY	BCID	CMPNT	CP	NRAD	XLOC	YLOC	ZLOC		
BODY	10	FUSEL	0	3					

Field	Contents
BCID	Body component identification number (Integer > 0)
CMPNT	Component type (FUSEL for the fuselage and POD for a pod)
CP	Coordinate system of the geometry input (Integer ≥ 0, or blank)
NRAD	Number of equal radial cuts used to define the body (Integer ≥ 0, or blank)
XLOC, YLOC, ZLOC	Ordinates of the nose of the pod in the CP coordinate system (Real)

Remarks:

1. NRAD is input if equally spaced radial cuts are desired. Arbitrary radial cuts are specified using the AXSTA and AEFAC data entries.
2. The geometry given with the XLOC, YLOC, ZLOC entries is used only with POD components.

Input Data Entry: CAERO1 Aerodynamic Panel Element Connection

Description: Defines an aerodynamic macroelement (panel) in terms of two leading edge locations and side chords. This is used for Doublet-Lattice theory.

Format and Example:

1	2	3	4	5	6	7	8	9	10
CAERO1	EID	PID	CP	NSPAN	NCHORD	LSPAN	LCHORD	IGID	CONT
+BC	X1	Y1	Z1	X12	X4	Y4	Z4	X43	

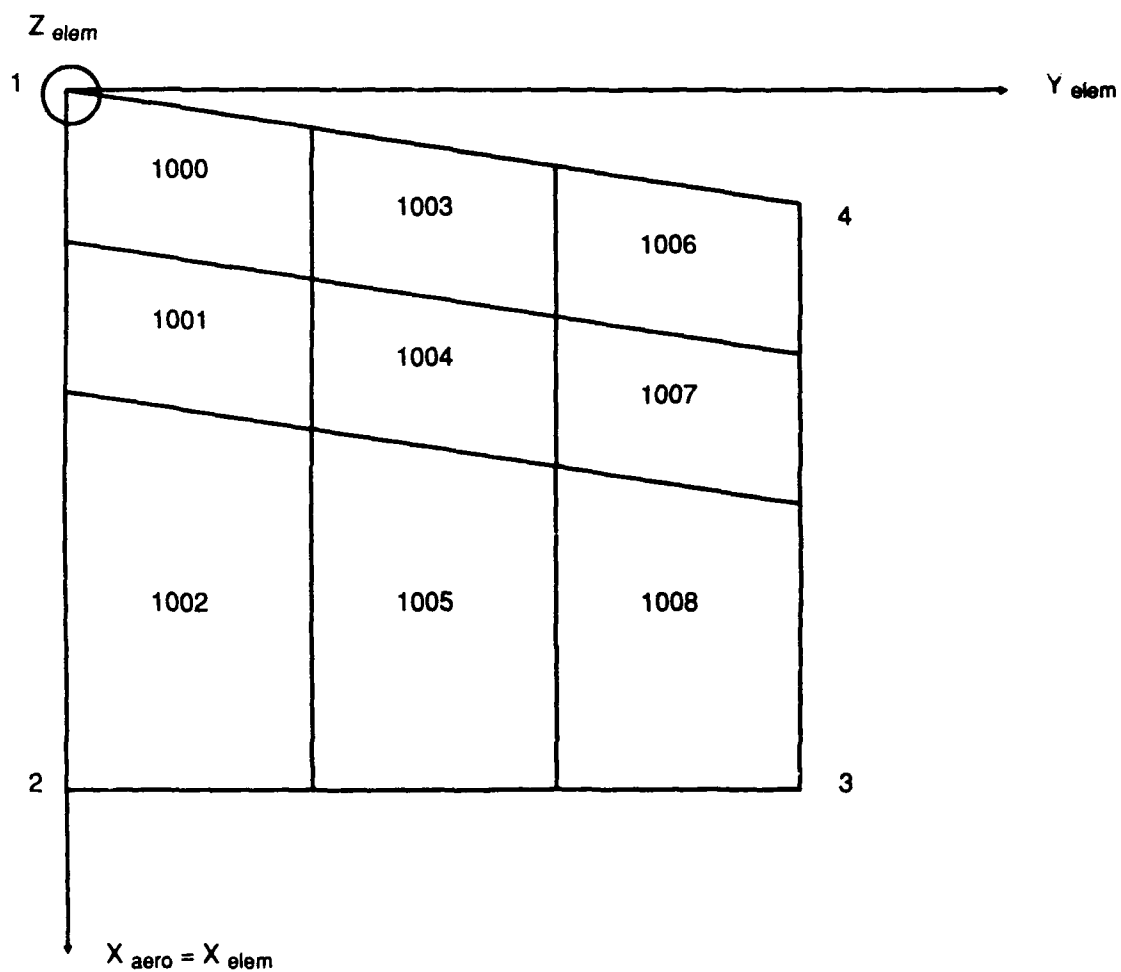
CAERO1	1000	1		3			2	1	ABC
+BC	0.0								

Field	Contents
EID	Element identification number (Integer > 0)
PID	Identification number of property entry (Integer > 0, or blank). Used to specify associated bodies
CP	Coordinate system for locating points 1 and 4 (Integer ≥ 0 or blank)
NSPAN	Number of span-wise boxes; if a positive value is given NSPAN , equal divisions are assumed; if zero or blank, a list of division points is given at LSPAN (Integer ≥ 0 or blank)
NCHORD	Number of chord-wise boxes; if a positive value is given NCHORD , equal divisions are assumed; if zero or blank, a list of division points is given at LCHORD (Integer ≥ 0 or blank)
LSPAN	Identification number of an AEFACT data entry containing a list of division points for span-wise boxes. Used only if NSPAN is zero or blank (Integer ≥ 0 or blank)
LCHORD	Identification number of an AEFACT data entry containing a list of division points for chord-wise boxes. Used only if NCHORD is zero or blank (Integer ≥ 0 or blank)
IGID	Interference group identification (aerodynamic elements with different IGIDs are uncoupled) (Integer > 0)
X1, Y1, Z1; X4, Y4, Z4;	Location of points 1 and 4, in coordinate system CP (Real)
X12; X43	Edge chord lengths (in aerodynamic coordinate system) (Real ≥ 0 , and not both zero)

Remarks:

1. The boxes are numbered sequentially, beginning with **EID**.
2. The continuation entry is required.
3. The number of division points is one greater than the number of boxes. Thus, if **NSPAN** = 3, the division points are 0.0, 0.333, 0.667, 1.000. If the user supplies division points, the first and last points need not be 0. and 1. (in which the corners to the panel would not be at the reference points).
4. A triangular element is formed if **X12** or **X43** = 0.0.

5. The element coordinate system (right-handed) is shown in the sketch below.



Input Data Entry: **CAERO2** Unsteady Aerodynamic Body Connection

Description: Defines an aerodynamic body for Doublet-Lattice aerodynamics.

Format and Examples:

1	2	3	4	5	6	7	8	9	10
CAERO2	EID	PID	CP	NSB	NINT	LSB	LINT	IGID	CONT
+BC	X1	Y1	Z1						

CAERO2	1500	2	100		4	99		1	ABC
+BC	-1.0	100	-30	175					

Field	Contents
EID	Element identification number (Integer > 0)
PID	Property identification number (Integer > 0)
CP	Coordinate system for locating point 1 (Integer ≥ 0, or blank)
NSB	Grid point identification number of connection points (Integer > 0)
NINT	Number of interference elements; if a positive number is given, NSB equal divisions are assumed; if zero or blank, see LSB for a list of divisions (Integer ≥ 0, or blank)
LSB	Identification number of an AEFAC data entry for slender body division points; used only if NSB is zero or blank (Integer ≥ 0 or blank)
LINT	Identification number of an AEFAC data entry containing a list of division points for interference elements; used only if NINT is zero or blank (Integer ≥ 0, or blank)
IGID	Interference group identification (aerodynamic elements with different IGID's are uncoupled) (Integer > 0)
X1, Y1, Z1	Location of points 1 and 4, in coordinate system CP (Real)
X12	Edge chord lengths (in aerodynamic coordinate system) (Real ≥ 0, and not both zero)

Remarks:

1. Point 1 is the leading point of the body.
2. All CAERO1 (panels) and CAERO2 (bodies) in the same group (IGID) will have aerodynamic interaction.
3. At least one interference element is required for each aerodynamic body specified by this entry.
4. Element identification numbers on the aerodynamic bodies must have the following sequence:
 - (A) Panels first
 - (B) Z bodies (see PAERO2 orientation flag)
 - (C) ZY bodies
 - (D) Y bodies

Input Data Entry: CAERO6

Description: Defines an aerodynamic macroelement (panel) for USSAERO.

Format and Example:

1	2	3	4	5	6	7	8	9	10
CAERO6	ACID	CMPNT	CP	IGRP	LCHORD	LSPAN			
CAERO6	1	WING		1	20	30			

Field	Contents
ACID	Component identification number (Integer > 0)
CMPNT	Aircraft component (Text)
CP	Coordinate system (Integer ≥ 0, or blank) (See Remark 5)
IGRP	Group number for this component (Integer > 0)
LCHORD	Identification number of an AEFACT Bulk Data entry containing a list of division points in percent chord for chord-wise boxes for the aerodynamic surface. If LCHORD is zero, the chord-wise divisions are identified by the IPANEL entry on the AIRFOIL Bulk Data entry (Integer ≥ ,0 or blank)
LSPAN	Identification number of an AEFACT Bulk Data entry containing a list of division points for spanwise boxes. For WINGS and CANARDS use the y (lateral) dimensional coordinates of the stations, and for FINS , use the z (vertical) dimensional coordinates. If LSPAN is zero or blank, the y/z locations from the AIRFOIL Bulk Data entries for the component ACID are used (Integer ≥ ,0 or blank)

Remarks:

1. Allowable components are **WING**, **FIN**, and **CANARD**.
2. The **IGRP** field allows related components to be processed together for interference effects; e.g., one group could be a wing/body/tail combination while a second group could be a pod/fin combination.
3. Note that the chord-wise cuts are in percent while the span-wise cuts require physical coordinates. For span-wise cuts, y-coordinates are input for wings and canards while z-coordinates are input for fins.
4. Only the right half-plane can be modeled in USSAERO. As such, all y-coordinates specified by **LSPAN** must be positive.
5. The basic coordinate system must be used (CP blank). This field exists for the addition of a user defined coordinate system in the future.

Input Data Entry: CBAR Simple Beam Element Connection

Description: Defines a simple beam element (BAR) of the structural model.

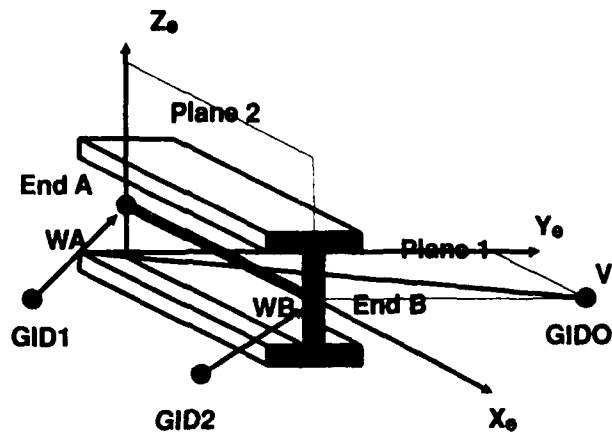
Format and Example:

1	2	3	4	5	6	7	8	9	10
CBAR	EID	PID	GA	GB	X1,GO	X2	X3	TMAX	CONT
CONT	PA	PB	W1A	W2A	W3A	W1B	W2B	W3B	
CBAR	2	39	7	3	13				
+23	123								

Field	Contents
EID	Unique element identification number (Integer > 0).
PID	Identification number of a PBAR property entry (Default is EID unless BAROR entry has nonzero entry in Field 3) (Integer > 0)
GA,GB	Grid point identification numbers of connection points (Integer > 0).
Xi	Components of vector {v}, at end A, measured at end A, parallel to the components of the displacement coordinate system for GA , to determine (with the vector from end A to end B) the orientation of the element coordinate system for the BAR element (Real)
GO	Grid point identification number to optionally supply xi (Integer > 0). Direction of orientation vector is GA to GO
TMAX	Maximum allowable cross-sectional area in design (Real > 0.0, or blank). Default = 10^4 .
PA, PB	Pin flags for bar ends A and B, respectively (up to 5 of the unique digits 1 through 6 anywhere in the fields with no embedded blanks; Integer > 0 or blank). Used to remove connections between the grid point and selected degrees of freedom of the bar. The degrees of freedom are defined in the element's coordinate system. The bar must have stiffness associated with the pin flag. For example, if PA=4 is specified, the PBAR entry must have a value for J , the torsional stiffness.
W1A,W2A,W3A W1B,W2B,W3B	Components of offset vectors wa and wb , respectively, in displacement coordinate systems at points GA and GB , respectively (Real or blank).

Remarks:

1. The element coordinate system is shown in the following figure:



2. If there are no pin flags or offsets, the continuation entry may be omitted.
3. The **TMAX** value is used only for shape function design variable linking.
4. See the **BAROR** entry for default options for Fields 3 and 6 through 8.

Input Data Entry: **CELAS1** Scalar Spring Connection

Description: Defines a scalar spring element of the structural model

Format and Example:

1	2	3	4	5	6	7	8	9	10
CELAS1	EID	PID	G1	C1	G2	C2	TMAX		
CELAS1	2	6			8	1			

Field	Contents
EID	Element identification number (Integer > 0)
PID	Identification number of a PELAS property entry (Default is EID) (Integer > 0)
G _i	Geometric grid point identification number (Integer ≥ 0)
C _i	Component number (6 ≥ Integer ≥ 0)
TMAX	Maximum value for design (Real, Default = 1.0 E4)

Remarks:

1. Scalar points may be used for G1 and/or G2 in which case the corresponding C1 and/or C2 must be zero or blank. Zero or blank may be used to indicate a grounded terminal G1 or G2 with a corresponding blank or zero C1 or C2. A grounded terminal is a point whose displacement is constrained to zero.
2. The two connection points (G1, C1) and (G2, C2) must be distinct.
3. **TMAX** is ignored unless the element is designed using shape function linking.

Input Data Entry: CELAS2 Scalar Spring Property and Connection

Description: Defines a scalar spring element of the structural model without reference to a property entry.

Format and Example:

1	2	3	4	5	6	7	8	9	10
CELAS2	EID	K	G1	C1	G2	C2	GE	S	CONT
CONT	TMIN	TMAX							
CELAS2	28	6.2+3	32		19	4			

Field	Contents
EID	Element identification number (Integer > 0)
K	The value of the scalar spring (Real > 0.0)
G1	Geometric grid point identification number (Integer ≥ 0)
C1	Component number (6 ≥ Integer ≥ 0)
GE	Damping coefficient (Real ≥ 0.0)
S	Stress coefficient (Real ≥ 0.0)
TMIN, TMAX	Minimum and maximum values for design (Real)

Remarks:

1. Scalar points may be used for G1 and/or G2 in which case the corresponding C1 and/or C2 must be zero or blank. Zero or blank may be used to indicate a grounded terminal G1 or G2 with a corresponding blank or zero C1 or C2. A grounded terminal is a point whose displacement is constrained to zero.
2. This single entry completely defines the element since no material or geometric properties are required.
3. The two connection points (G1, C1) and (G2, C2) must be distinct.
4. The TMIN and TMAX values are ignored unless shape function design variable linking is used.

Input Data Entry: **CIHEX1** Linear Isoparametric Hexahedron Element Connection

Description: Defines a linear isoparametric hexahedron element of the structural model.

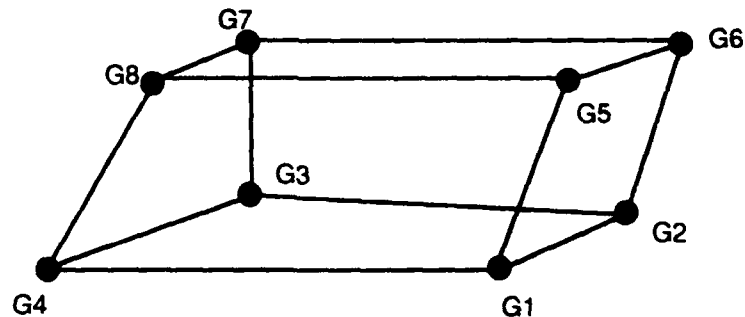
Format and Example:

1	2	3	4	5	6	7	8	9	10
CIHEX1	EID	PID	G1	G2	G3	G4	G5	G6	CONT
CONT	G7	G8							
CIHEX1	137	5	3	8	5	4	9	14	ABC
+BC	88	602							

Field	Contents
EID	Element identification number (Integer > 0).
PID	Identification number of a PIHEX property entry (Integer > 0). (Default is EID)
Gi	Grid point identification numbers of connection points (Integer > 0, G1 ≠ G2 ≠ ...G8).

Remarks:

1. Grid points G1, G2, G3, and G4 must be given in counterclockwise order about one quadrilateral, with G1 and G5 along the same edge as shown in the figure below:



2. There is no nonstructural mass.
3. The quadrilateral faces need not be planar.
4. Stresses are given in the basic coordinate system.
5. The continuation is required.
6. No physical property in this element can be used as a local design variable for automated design.

Input Data Entry: CIHEX2 Quadratic Isoparametric Hexahedron Element Connection

Description: Defines a quadratic isoparametric hexahedron element of the structural model.

Format and Example:

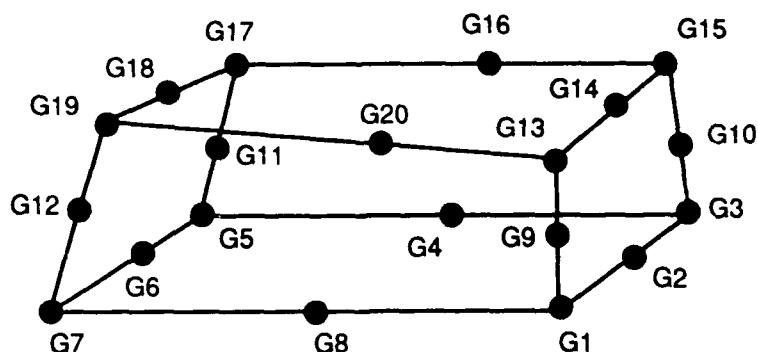
1	2	3	4	5	6	7	8	9	10
CIHEX1	EID	PID	G1	G2	G3	G4	G5	G6	CONT
CONT	G7	G8	G9	G10	G11	G12	G13	G14	CONT
CONT	G15	G16	G17	G18	G19	G20			

CIHEX1	110	7	3	8	12	13	14	9	ABC
+BC	5	4	16	19	20	17	23	27	DEF
+EF	31	32	33	28	25	24			

Field	Contents
EID	Element identification number (Integer > 0)
PID	Identification number of a PIHEX property entry (Integer > 0) (Default is EID)
Gi	Grid point identification numbers of connection points (Integer > 0, G1 ≠ G2 ≠ ... ≠ G20).

Remarks:

1. Grid points G1,...,G8 must be given in counterclockwise order about one quadrilateral face when viewed from inside the element. G9,...,G12 and G13,...,G20 must also be in a counterclockwise direction with G1, G9 and G13 along the same edge as shown in the figure below:



2. There is no nonstructural mass.
3. The quadrilateral faces need not be planar.
4. Stresses are given in the basic coordinate system.
5. The continuations are required.
6. No physical property in this element can be used as a local design variable for automated design.

Input Data Entry: **CIHEX3** Cubic Isoparametric Hexahedron Element Connection

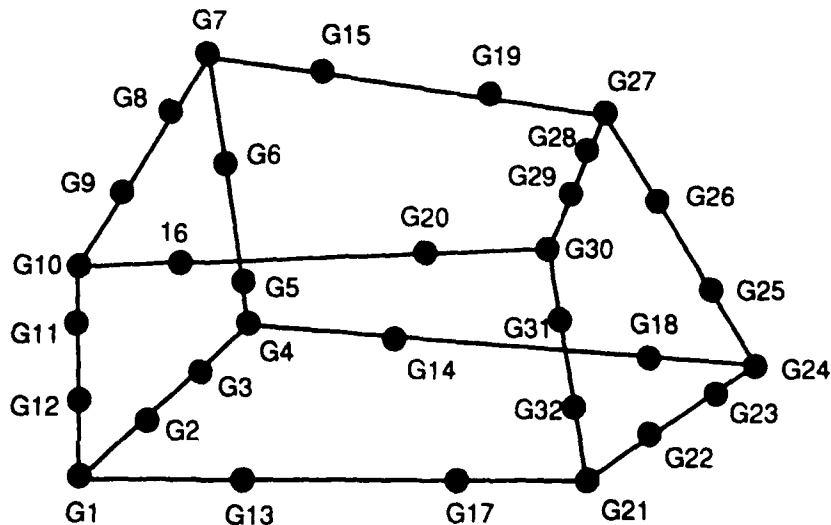
Description: Defines a cubic isoparametric hexahedron element of the structural model.

Format and Example:

1	2	3	4	5	6	7	8	9	10
CIHEX1	EID	PID	G1	G2	G3	G4	G5	G6	CONT
CONT	G7	G8	G9	G10	G11	G12	G13	G14	CONT
CONT	G15	G16	G17	G18	G19	G20	G21	G22	CONT
CONT	G23	G24	G25	G26	G27	G28	G29	G30	CONT
CONT	G31	G32							

CIHEX1	15	3	4	9	12	17	18	19	ABC
+BC	20	13	10	7	6	5	22	25	DEF
+EF	31	32	33	28	25	24	108	214	GHI
+HI	106	213	413	95	67	40	45	90	+KL
+KL	38	37							

Field	Contents
EID	Element identification number (Integer > 0).
PID	Identification number of a PIHEX property entry (Integer > 0) (Default is EID)
Gi	Grid point identification number of connection points (Integer > 0, G1 ≠ G2 ≠ ... ≠ G32).



Remarks:

1. Grid points G_1, \dots, G_{12} must be given in counterclockwise order about one quadrilateral face when viewed from inside the element. G_{13}, \dots, G_{16} ; G_{17}, \dots, G_{20} ; and G_{21}, \dots, G_{32} must also be in a counterclockwise direction with G_1 , G_{13} , G_{17} , and G_{21} along the same edge.
2. There is no nonstructural mass.
3. The quadrilateral faces need not be planar.
4. Stresses are given in the basic coordinate system.
5. The continuations are required.
6. No physical property in this element can be used as a local design for automated design.

Input Data Entry: **CMASS1** Scalar Mass Connection

Description: Defines a scalar mass element of the structural model.

Format and Example:

1	2	3	4	5	6	7	8	9	10
CMASS1	EID	PID	G1	C1	G2	C2	TMAX		
CMASS1	32	6	2	1					

Field	Contents
EID	Element identification number (Integer > 0)
PID	Identification number of a PMASS property entry (Default is EID) (Integer > 0)
Gi	Geometric grid point identification number (Integer > 0)
Ci	Component number ($6 \geq \text{Integer} \geq 0$)
TMAX	The maximum mass value allowed in design (Real, Default = 10^4)

Remarks:

1. Scalar points may be used for G1 and/or G2 in which case the corresponding C1 and/or C2 must be zero or blank. Zero or blank may be used to indicate a grounded terminal G1 or G2 with a corresponding blank or zero C1 or C2. A grounded terminal is a point whose displacement is constrained to zero.
2. The two connection points (G1, C1) and (G2, C2), must be distinct. Except in unusual circumstances, one of them will be a grounded terminal with blank entries for G and C.
3. The **TMAX** value is used only for shape function design variable linking.

Input Data Entry: CMASS2 Scalar Mass Property and Connection

Description: Defines a scalar mass element of the structural model without reference to a property entry.

Format and Example:

1	2	3	4	5	6	7	8	9	10
CMASS2	EID	M	G1	C1	G2	C2	TMIN	TMAX	
CMASS2	32	9.25	6	1					

Field	Contents
EID	Element identification number (Integer > 0)
M	The value of the scalar mass (Real)
G1	Geometric grid point identification number (Integer > 0)
C1	Component number $6 \geq \text{Integer} \geq 0$
TMIN, TMAX	The minimum and maximum mass values in design (Real)

Remarks:

1. Scalar points may be used for G1 and/or G2 in which case the corresponding C1 and/or C2 must be zero or blank. Zero or blank may be used to indicate a grounded terminal G1 or G2 with a corresponding blank or zero C1 or C2. A grounded terminal is a point whose displacement is constrained to zero.
2. This single entry completely defines the element since no material or geometric properties are required.
3. The two connection points (G1, C1) and (G2, C2), must be distinct. Except in unusual circumstances, one of them will be a grounded terminal with blank entries for G and C.
4. The TMIN and TMAX values are used only for shape function design variable linking.

Input Data Entry: **CONEFFF** Flutter aerodynamic control effectiveness data

Description: Defines adjustment factors of control surface effectiveness values for use in flutter analysis.

Format and Example:

1	2	3	4	5	6	7	8	9	10
CONEFFF	EFFID	EFF	MODE	MACROID	BOX1	BOX2			
CONEFFF	10	0.60	6	1001	1007	1021			

Field	Contents
EFFID	Effectiveness identification number (Integer > 0)
EFF	Effectiveness value (Real)
MODE	Structural mode to which the effectiveness is to be applied (Integer > 0)
MACROID	Aerodynamic component (macroelement) on which the control surface lies
BOX1, BOX2	First and last box whose effectiveness is to be altered (Integer > 0, BOX2 > BOX1)

Remarks:

1. The **EFFID** is referenced by the **FLUTTER** bulk data entry.
2. The **EFFID** need not be unique.
3. The pressures for the referenced mode and all the referenced boxes will be modified by the **EFF** parameter. For example, **EFF** = 0.60 indicates a 40 percent reduction in the effectiveness for the affected boxes.
4. Refer to the **SPLINE1** bulk data entry for the interpretation of **BOX1** and **BOX2**.

Input Data Entry: **CONEFFS** Static aerodynamic control effectiveness data

Description: Defines adjustment factors for control surface effectiveness values for use in static aeroelastic analysis and nonplanar aerodynamic analysis.

Format and Example:

1	2	3	4	5	6	7	8	9	10
CONEFFS	EFFID	LABEL1	EFF1	LABEL2	EFF2	LABEL3	EFF3		CONT
CONT	LABEL4	EFF4	-etc-						
CONEFFS	10	AIL1	0.65	INBORD	0.55				

Field	Contents
EFFID	A unique identification number identifying the set
LABELi	A unique alphanumeric string of up to eight characters to identify a control surface defined by an AESURF entry
EFFi	Effectiveness value for the associated surface (Real)

Remarks:

1. The set identification number is referenced by the **TRIM** bulk data entry.
2. All aerodynamic forces created by the control surface will be reduced to the reference amount. For example, **EFF1** = 0.70 indicates a 30 percent reduction in the forces.

Input Data Entry: **CONLINK** Linked Control Surfaces

Description: Causes control surfaces to vary in a prescribed fashion relative to one another.

Format and Example:

1	2	3	4	5	6	7	8	9	10
CONLINK	LABEL		LABEL1	VAL1	LABEL2	VAL2	LABEL3	VAL3	CONT
CONT	LABEL4	VAL4	-etc-						

CONLINK	ROLL1		AIL	1.0	LEFLAP	1.0			
---------	-------	--	-----	-----	--------	-----	--	--	--

Field	Contents
LABEL	A unique alphanumeric string of up to eight characters to identify the control surface taht is composed of other control surfaces.
LABELi	A unique alphanumeric string of up to eight characters to identify a control surface defined by an AESURF entry
VALi	Participation factor (Real)

Remarks:

1. All of the **LABEL** surfaces must be of the same **TYPE**, e.g. **SYM**. See the **AESURF** entry for additional information.
2. An arbitrary number of entries are allowed.
3. The **CONLINK** entry may not reference the **LABEL** of another **CONLINK** entry.

Input Data Entry: **CONM1** Concentrated Mass Element Connection, General Form

Description: Defines a 6 x 6 symmetric matrix at a geometric grid point of the structural model.

Format and Example:

1	2	3	4	5	6	7	8	9	10
CONM1	EID	G	CID	M11	M21	M22	M31	M32	CONT
CONT	M33	M41	M42	M43	M44	M51	M52	M53	CONT
CONT	M54	M55	M61	M62	M63	M64	M65	M66	

CONM1	2	22	2	2.9	6.3	+1			+1
-1	4.8				28.6				+2
+2		28.6						28.6	

Field	Contents
EID	Element identification number (Integer > 0).
G	Grid point identification number (Integer > 0).
CID	Coordinate system identification number for the mass matrix (Integer ≥ 0 or blank)
Mij	Mass matrix values (Real).

Remarks:

1. For a less general means of defining concentrated mass at grid points, see **CONM2**.
2. No physical property in this element can be used as a local design variable for automated design.

Input Data Entry: **CONM2** Concentrated Mass Element Connection, Rigid Body Form

Description: Defines a concentrated mass at a grid point of the structural model.

Format and Example:

1	2	3	4	5	6	7	8	9	10
CONM2	EID	G	CID	M	X1	X2	X3		CONT
CONT	I11	I21	I22	I31	I32	I33	TMIN	TMAX	

CONM2	2	15	6	49.7					123
+23	16.2		16.2			7.8			

Field	Contents
EID	Element identification number (Integer > 0).
G	Grid point identification number (Integer > 0).
CID	Coordinate system identification number (Integer ≥ -1). A CID of -1 (integer) allows the user to input x1 as the center of gravity location in the basic coordinate system. A CID of 0 implies the basic coordinate system
M	Mass value (Real).
Xi	Offset distances from the grid point to the center of gravity of the mass in the coordinate system defined in Field 4, unless CID = -1, in which case xi are the coordinates of the center of gravity of the mass in the basic coordinate system (Real).
Iij	Mass moments of inertia measured at the mass c.g., in coordinate system defined by Field 4 (Real). If CID = -1, the basic coordinate system is implied.
TMIN, TMAX	The minimum and maximum mass values for design (Real)

Remarks:

1. The continuation entry may be omitted.
2. If CID = -1, offsets are internally computed as the difference between the grid point location and **xi**. The grid point locations may be defined in a nonbasic coordinate system. In this case, the values of **Iij** must be in a coordinate system that parallels the basic coordinate system.

3. The form of the inertia matrix about its c.g. is taken as:

$$M = \begin{bmatrix} M & & & & \\ & M & & & \\ & & M & & \\ & & & I11 & \\ & & & -I21 & I22 \\ & & & -I31 & -I32 & I33 \end{bmatrix}$$

where $M = \int p v d$

$$I11 = \int p (x_2^2 + x_3^2) dv$$

$$I22 = \int p (x_1^2 + x_3^2) dv$$

$$I33 = \int p (x_1^2 + x_2^2) dv$$

$$I21 = \int p x_1 x_2 dv$$

$$I31 = \int p x_1 x_3 dv$$

$$I32 = \int p x_2 x_3 dv$$

and x_1, x_2, x_3 are components of distance from the c.g. in the coordinate system defined in Field 4. The negative signs for the off-diagonal terms are supplied by the program. A warning message is issued if the inertia matrix is non-positive definite, as this may cause fatal errors in dynamic analysis modules.

4. For design, the mass moments of inertia must be zero.
5. The **TMIN** and **TMAX** values are used only for shape function design variable linking.

Input Data Entry: CONROD Rod Element Property and Connection

Description: Defines a rod element of the structural model without reference to a property entry.

Format and Example:

1	2	3	4	5	6	7	8	9	10
CONROD	EID	G1	G2	MID	A	J	C	NSM	CONT
CONT	TMIN	TMAX							
CONROD	2	16	17	23	2.69				

Field	Contents
EID	Element identification number (Integer > 0).
G1, G2	Grid point identification numbers of connection points (Integer > 0)
MID	Material identification number (Integer > 0).
A	Area of rod (Real ≥ 0.0).
J	Torsional constant (Real ≥ 0.0).
C	Coefficient for torsional stress determination (Real).
NSM	Nonstructural mass per unit length (Real).
TMIN, TMAX	Minimum and maximum allowable cross-sectional areas in design (Real > 0.0, or blank)

Remarks:

1. For structural problems, CONROD entries may only reference MAT1 material entries.
2. The continuation entry is optional.
3. TMAX and TMIN are ignored unless element is linked to global design variable through a SHAPE entry.

Input Data Entry: CONVERT

Description: Defines conversion factors for various physical quantities.

Format and Example:

1	2	3	4	5	6	7	8	9	10
CONVERT	QUANT1	FACTOR	QUANT2	FACTOR	QUANT5	FACTOR	QUANT4	FACTOR	CONT
CONT	QUANT	FACTOR	QUANT	FACTOR	-etc-				
CONVERT	MASS	0.00259							

Field	Contents
QUANTi	A character string identifying the physical quantity to be converted = MASS , or VELOCITY
FACTOR	The conversion factor (Real \neq 0.0)

Remarks:

1. Any number of valid quantity-factor combinations can be entered on a single entry.
2. Only **MASS** and **VELOCITY** are currently valid quantity entries.
3. Input mass values will be multiplied by the input factor. Input velocities will be multiplied by the factor.

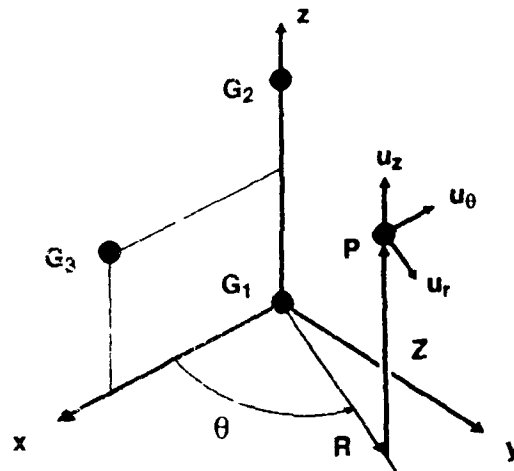
Input Data Entry: CORD1C Cylindrical Coordinate System Definition, Form 1.

Description: Defines a cylindrical coordinate system by reference to three grid points. These points must be defined in coordinate systems whose definition does not involve the coordinate system being defined. The first point is the origin, the second lies on the z-axis, and the third lies in the plane of the azimuthal origin.

Format and Example:

1	2	3	4	5	6	7	8	9	10
CORD1C	CID	G1	G2	G3	CID	G1	G2	G3	
CORD1C	3	16	32	19					

Field	Contents
CID	Coordinate system identification number (Integer > 0)
G1	Grid point identification number (Integer > 0; $G1 \neq G2 \neq G3$).



Remarks:

1. Coordinate system identification numbers on all CORD1R, CORD1C, CORD1S, CORD2R, CORD2C, and CORD2S entries must be unique.
2. The three points G1, G2, and G3 must be noncollinear.
3. The location of a grid point (P in the sketch) in this coordinate system is given by (R, θ, Z) where θ is measured in degrees.
4. The displacement coordinate directions at P are dependent on the location of P as shown above by (u_r, u_θ, u_z) .
5. Points on the z-axis may not have their displacement directions defined in this coordinate system since an ambiguity results.
6. One or two coordinate systems may be defined on a single entry.

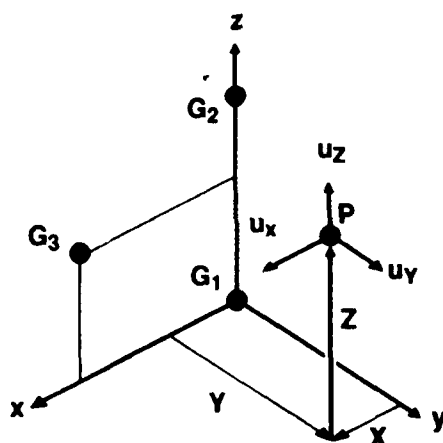
Input Data Entry: CORD1R Rectangular Coordinate System Definition, Form 1.

Description: Defines a rectangular coordinate system by reference to three grid points. These points must be defined in coordinate systems whose definition does not involve the coordinate systems defined. The first point is the origin, the second lies on the z-axis, and the third lies in the x-z plane.

Format and Example:

1	2	3	4	5	6	7	8	9	10
CORD1R	CID	G1	G2	G3	CID	G1	G2	G3	
CORD1R	3	16	32	19					

Field	Contents
CID	Coordinate system identification number (Integer > 0)
Gi	Grid point identification number (Integer > 0; $G1 \neq G2 \neq G3$).



Remarks:

1. Coordinate system identification numbers on all CORD1R, CORD1C, CORD1S, CORD2R, CORD2C, and CORD2S entries must be unique.
2. The three points G1, G2, and G3 must be noncollinear.
3. The location of a grid point (P in the sketch) in this coordinate system is given by (X, Y, Z).
4. The displacement coordinate directions at P are shown above by (u_x , u_y , u_z).
5. One or two coordinate systems may be defined on a single entry.

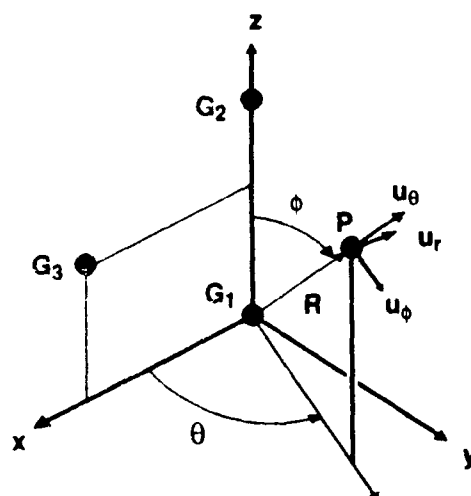
Input Data Entry: CORD1S Spherical Coordinate System Definition, Form 1.

Description: Defines a spherical coordinate system by reference to three grid points. These points must be defined in coordinate systems whose definition does not involve the coordinate systems defined. The first point is the origin, the second lies on the z-axis, and the third lies in the plane of the azimuthal origin.

Format and Examples:

1	2	3	4	5	6	7	8	9	10
CORD1S	CID	G1	G2	G3	CID	G1	G2	G3	
CORD1S	0	16	32	19					

Field	Contents
CID	Coordinate system identification number (Integer > 0)
G1	Grid point identification number (Integer > 0; $G1 \neq G2 \neq G3$).



Remarks:

1. Coordinate system identification numbers on all CORD1R, CORD1C, CORD1S, CORD2R, CORD2C, and CORD2S entries must be unique.
2. The three points G1, G2, and G3 must be noncollinear.
3. The location of a grid point (P in the sketch) in this coordinate system is given by (R, θ, ϕ) where θ and ϕ are measured in degrees.
4. The displacement coordinate directions at P are dependent on the locations of P as shown above by (u_r, u_θ, u_ϕ) .
5. Points in the polar axis may not have their displacement direction defined in this coordinate system since an ambiguity results.
6. One or two coordinate systems may be defined on a single entry.

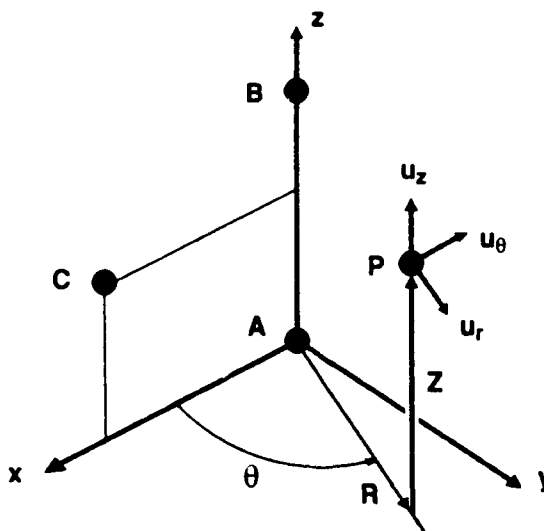
Input Data Entry: CORD2C Cylindrical Coordinate System Definition, Form 2.

Description: Defines a cylindrical coordinate system by reference to the coordinates of three grid points. The first point defines the origin. The second point defines the direction of the z-axis. The third lies in the plane of the azimuthal origin. The reference coordinate system must be independently defined.

Format and Example:

1	2	3	4	5	6	7	8	9	10
CORD2C	CID	RID	A1	A2	A3	B1	B2	B3	CONT
CONT	C1	C2	C3						
CORD2C	3	17	-2.9	1.0	0 0	3.6	0.0	1.0	123
+23	5.2	1.0	-2.9						

Field	Contents
CID	Coordinate system identification number (Integer > 0)
RID	Reference to a coordinate system which is defined independently of new coordinate system (Integer ≥ 0, or blank)
A1, B1, C1	Coordinates of three points in coordinate system defined in Field 3 (Real)



Remarks:

1. Continuation entry must be present.
2. The three points (A1, A2, A3), (B1, B2, B3), (C1, C2, C3) must be unique and noncollinear. Noncollinearity is checked by the geometry processor.
3. Coordinate system identification numbers on all CORD1R, CORD1C, CORD1S, CORD2R, CORD2C, and CORD2S entries must all be unique.

4. An RID of zero references the basic ordinate system.
5. The location of a grid point (P in the sketch) in this coordinate is given by (R, θ, Z) where θ is measured in degrees.
6. The displacement coordinate directions at P are dependent on the location of P as shown above by (u_r, u_θ, u_z) .
7. Points on the z-axis may not have their displacement direction defined in this coordinate system since an ambiguity results.

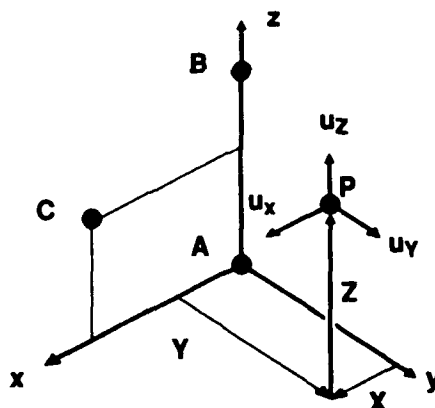
Input Data Entry: CORD2R Rectangular Coordinate System Definition, Form 2.

Description: Defines a rectangular coordinate system by reference to coordinates of three points. The first point defines the origin. The second defines the direction of the z-axis. The third point defines a vector which, with the z-axis, defines the x-z plane. The reference coordinate system must be independently defined.

Format and Example:

1	2	3	4	5	6	7	8	9	10
CORD2R	CID	RID	A1	A2	A3	B1	B2	B3	CONT
CONT	C1	C2	C3						
CORD2R	3	17	-2.9	1.0	0.0	3.6	0.0	1.0	123
+23	5.2	1.0	-2.9						

Field	Contents
CID	Coordinate system identification number (Integer > 0)
RID	Reference to a coordinate system which is defined independently of new coordinate system (Integer ≥ 0, or blank)
Ai, Bi, Ci	Coordinates of three points in coordinate system defined in Field 3 (Real)



Remarks:

1. The continuation entry must be present.
2. The three points (A1, A2, A3), (B1, B2, B3), (C1, C2, C3) must be unique and noncollinear. Noncollinearity is checked by the geometry processor.
3. Coordinate system identification numbers on all CORD1R, CORD1C, CORD1S, CORD2R, CORD2C, and CORD2S entries must all be unique.
4. An RID of zero references the basic coordinate system.
5. The location of a grid point (P in the sketch) in this coordinate system is given by (X, Y, Z)
6. The displacement coordinate directions at P are shown by (u_x , u_y , u_z)

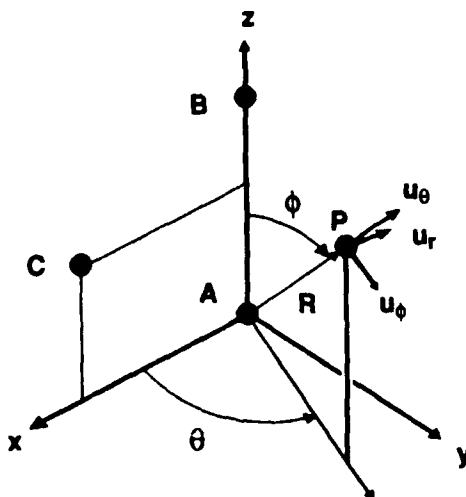
Input Data Entry: CORD2S Spherical Coordinate System Definition, Form 2.

Description: Defines a spherical coordinate system by reference to the coordinates of three points. The first point defines the origin. The second point defines the direction of the z-axis. The third lies in the plane of the azimuthal origin. The reference coordinate system must be independently defined.

Format and Example:

1	2	3	4	5	6	7	8	9	10
CORD2S	CID	RID	A1	A2	A3	B1	B2	B3	CONT
CONT	C1	C2	C3						
CORD2S	3	17	-2.9	1.0	0.0	3.6	0.0	1.0	123
+23	5.2	1.0	-2.9						

Field	Contents
CID	Coordinate system identification number (Integer > 0)
RID	Reference to a coordinate system which is defined independently of of new coordinate system (Integer ≥ 0, or blank)
Ai, Bi, Ci	Coordinates of three points in coordinate system defined in Field 3 (Real)



Remarks:

1. The continuation entry must be present.
2. The three points (A1, A2, A3), (B1, B2, B3), (C1, C2, C3) must be unique and noncollinear.
3. Coordinate system identification numbers on all CORD1R, CORD1C, CORD1S, CORD2R, CORD2C and CORD2S entries must all be unique.
4. An RID of zero references the basic coordinate system.

5. The location of a grid point (P in the sketch) in this coordinate system is given by (R, θ, ϕ) where θ , and ϕ are measured in degrees.
6. The displacement coordinate directions at P are shown above by (u_r, u_θ, u_ϕ) .
7. Points on the polar axis may not have their displacement directions defined in this coordinate system since an ambiguity results.

Input Data Entry: **CQDMEM1** Isoparametric Quadrilateral Element Connection

Description: Defines the isoparametric quadrilateral membrane element.

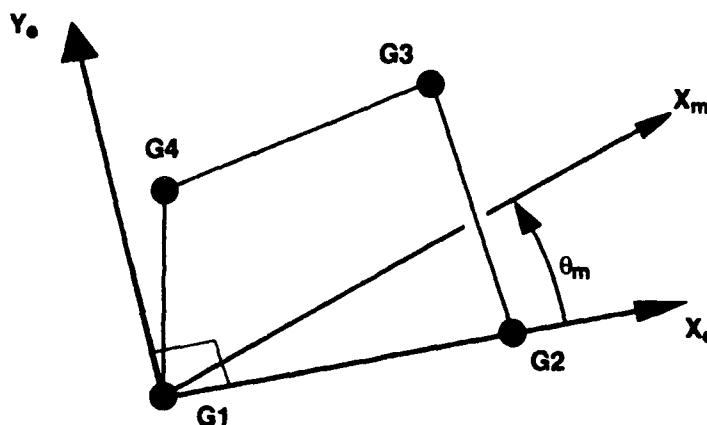
Format and Example:

1	2	3	4	5	6	7	8	9	10
CQDMEM1	EID	PID	G1	G2	G3	G4	TH	TMAX	
CQDMEM1	72	13	13	14	15	16	29.2		

Field	Contents
EID	Element identification number (Integer > 0).
PID	Identification number of a PQDMEM1 or PCOMP property entry (Default is EID) (Integer > 0).
G1	Grid point identification numbers of connection points (Integer > 0)
TH	Material property orientation angle. If TH is real, the sketch below gives the sign convention for TH . If TH is an integer, the material x-axis is along the x-axis of coordinate system identified by the integer.
TMAX	Maximum allowable element thickness in design (Real > 0.0 or blank). (Default=10 ⁴)

Remarks:

1. Grid points **G1** through **G4** must be ordered consecutively around the perimeter of the element as shown in the figure below.



2. All interior angles must be less than 180°.
3. **TMAX** is ignored unless element is linked to global design variable by a **SHAPE** entry.

Input Data Entry: CQUAD4 Quadrilateral Element Connection

Description: Quadrilateral plate element (QUAD4) of the structural model. This is an isoparametric membrane-bending element.

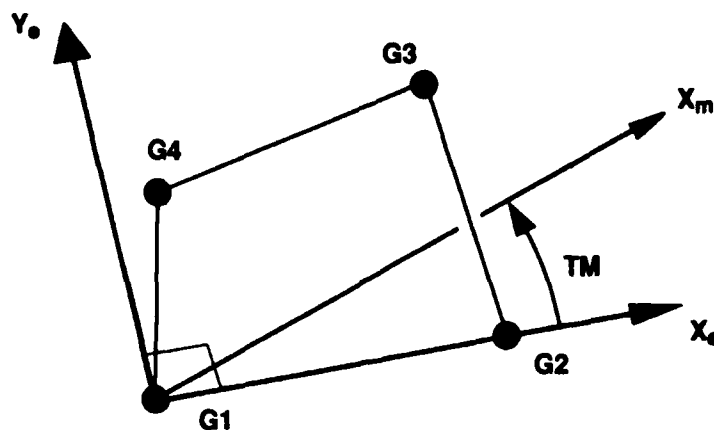
Format and Example:

1	2	3	4	5	6	7	8	9	10
CQUAD4	EID	PID	G1	G2	G3	G4	TM	ZOFF	CONT
CONT		TMAX	T1	T2	T3	T4			
CQUAD4	101	17	1001	1005	1010	1024	45.0	0.01	ABC
+BC			0.03	0.125	0.05	0.04			

Field	Contents
EID	Element identification number (Integer > 0)
PID	Identification number of a PSHELL or PCOMP1 entry (Default is EID) (Integer > 0).
Gi	Grid point identification numbers of connection points (Integer > 0).
ZOFF	Offset of the element reference plane from the plane of grid points. A positive value means the +z _e direction. (Real or blank, see Remark 2 for default).
TM	Material property orientation specification (Real or blank; or 0 ≤ Integer < 1,000,000). If Real or blank, specifies the material property orientation angle in degrees. If Integer, the orientation of the material x-axis is along the projection onto the plane of the element of the x-axis of the coordinate system specified by the integer value.
TMAX	Maximum allowable element thickness in design (Real > 0.0).
Ti	Membrane thickness of element at grid points Gi (Real or blank, see Remark 3 for default).

Remarks:

1. The QUAD4 geometry, coordinate systems and numbering are shown in the figure below:



2. The material coordinate system (**TM**) and the offset (**ZOFF**) may also be provided on the **PSHELL** entry. The property data will be used if the corresponding field on the **CQUAD4** entry is blank. The element reference plane is located at the mid-thickness of the element parallel to the element mean plane.
3. The **T1** are optional, if not supplied they will be set to the value of **T** specified on the **PSHELL** entry. In such cases, the continuation entry is not required.
4. **TMAX** is ignored unless the element is linked to the global design variables by a **SHAPE** entry.

Input Data Entry: **CROD** Rod Element Connection

Description: Defines a tension-compression-torsion element (**ROD**) of the structural model.

Format and Examples:

1	2	3	4	5	6	7	8	9	10
CROD	EID	PID	G1	G2	TMAX				
CROD	12	13	21	23					

Field	Contents
EID	Element identification number (Integer > 0).
PID	Identification number of a PROD property entry (Default is EID) (Integer > 0).
Gi	Grid point identification numbers of connection points (Integer > 0)
TMAX	Maximum allowable rod area in design (Real > 0.0 or blank)

Remarks:

1. See **CONROD** for alternative method of rod definition.
2. Only one **ROD** element may be defined on a single entry.
3. **TMAX** is ignored unless the element is linked to global design variables by a **SHAPE** entry.

Input Data Entry: **CSHEAR** Shear Panel Element Connection

Description: Defines a shear panel element (SHEAR) of the structural model.

Format and Example:

1	2	3	4	5	6	7	8	9	10
CSHEAR	EID	PID	G1	G2	G3	G4	TMAX		
CSHEAR	3	6	1	5	3	7			

Field	Contents
EID	Element identification number (Integer > 0).
PID	Identification number of a PSHEAR property entry (Default is EID) (Integer > 0).
G1	Grid point identification numbers of connection points (Integer > 0)
TMAX	Maximum allowable thickness in design (Real > 0.0, or blank).

Remarks:

1. Grid points G1 through G4 must be ordered consecutively around the perimeter of the element.
2. All interior angles must be less than 180°.
3. TMAX is ignored unless element is linked to global design variable by a SHAPE entry.

Input Data Entry: CTRIA3 Triangular Element Connection

Description: Defines a triangular shell element (TRIA3) of the structural model.

Format and Example:

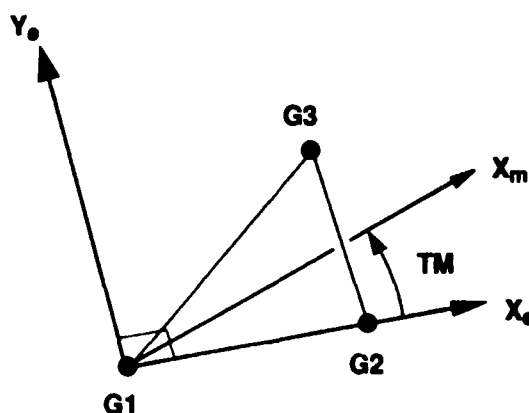
1	2	3	4	5	6	7	8	9	10
CTRIA3	EID	PID	G1	G2	G3	TM	ZOFF		abc
+BC			TMAX	T2	T3	T4			

CTRIA3	101	17	1001	1005	1010	45.0	0.01		ABC
+BC				0.03	0.125	0.05			

Field	Contents
EID	Element identification number (Integer > 0)
PID	Identification number of a PSHELL or PCOMP1 property entry (Default is EID) (Integer > 0).
Gi	Grid point identification numbers of connection points (Integer > 0).
ZOFF	Offset of the element reference plane from the plane of grid points. A positive value means the +z _e direction. (Real or blank, see Remark 2 for default).
TM	Material property orientation specification (Real or blank; or 0 ≤ Integer < 1,000,000). If Real or blank, specifies the material property orientation angle in degrees. If Integer, the orientation of the material x-axis is along the projection onto the plane of the element of the x-axis of the coordinate system specified by the integer value.
TMAX	Maximum allowable element thickness in design (Real > 0.0) (Default = 10 ⁴)
Ti	Membrane thickness of element at grid points G1 (Real or blank, see Remark 3 for default).

Remarks:

1. The TRIA3 geometry, coordinate systems and numbering are shown in the figure below:



2. The material coordinate system (TM) and the offset (ZOFF) may also be provided on the PSHELL entry. The property data will be used if the corresponding field on the CTRIA3 entry is blank. The element reference plane is located at the mid-thickness of the element parallel to the element mean plane.

3. The **T1** are optional, if not supplied they will be set to the value of **T** specified on the **PSHELL** entry. In such cases, the continuation entry is not required.
4. **TMAX** is ignored unless the element is linked to the global design variables by a **SHAPE** entry.

Input Data Entry: **CTRMEM**

Description: Defines a triangular membrane element.

Format and Examples:

1	2	3	4	5	6	7	8	9	10
CTRMEM	EID	PID	G1	G2	G3	THETA	TMAX		
CTRMEM	100	500	1	7	12				

Field	Contents
EID	Element identification number (Integer > 0).
PID	Identification of PTRMEM or PCOMP entry (Integer > 0) Default = EID .
Gi	Grid point identifications of connection points (Integer > 0).
THETA	Material orientation angle (Real) or $0 < \text{Integer} < 1,000,000$. If integer, then material x-axis lies along x-axis of coordinate system identified by the integer.
TMAX	Maximum allowable thickness in design. (Real ≥ 0 ., Default = 10^4)

Remarks:

1. The **TMAX** value is used only for shape function design variable linking.

Input Data Entry: DCONALE

Description: Defines an aileron effectiveness constraint of the form:

$$AE \leq \text{AEREQ (upper bound)} \text{ or } AE \geq \text{AEREQ (lower bound)}$$

where,

$$AE = \frac{-C_{\delta a}}{-C_{\delta b} \frac{pb}{2v}}$$

Format and Example:

1	2	3	4	5	6	7	8	9	10
DCONALE	SID	LABEL	CTYPE	AEREQ					
DCONALE	25	OUTBDAIL	LOWER	0.4					

Field	Contents
SID	Aerodynamic set identification for the imposed constraint (Integer > 0)
LABEL	A unique alphanumeric string of up to eight characters to identify the AESURF or CONLINK control surface
CTYPE	Constraint type: either UPPER for upper bound or LOWER for lower bound (Character, Default = LC:IER)
AEREQ	Required aileron effectiveness (Real \neq 0.0)

Remarks:

1. This constraint constraint will only be applied if selected by the Solution Control discipline option **DCON=SID** and if an antisymmetric aeroelastic trim analysis is being performed.
2. A **LOWER** bound constraint excludes all values to the left of **AEREQ** on a real number line, while an **UPPER** bound constraint excludes all values to the right, irrespective of the sign of **AEREQ**.
3. The effectiveness in roll of multiple control surfaces may be specified using multiple **DCONALE** entries with one constraint generated for each **LABEL/CTYPE** combination.

Input Data Entry: DCONCLA

Description: Defines a flexible lift curve slope constraint of the form:

$$CLA \leq CLAREQ \text{ or } CLA \geq CLAREQ$$

where,

$$CLA = \frac{(C_{l\alpha})_f}{(C_{l\alpha})_r}$$

Format and Example:

1	2	3	4	5	6	7	8	9	10
DCONCLA	SID	CTYPE	CLAREQ						
DCONCLA	25	UPPER	0.8						

Field	Contents
SID	Aerodynamic set identification for the imposed constraint (Integer > 0)
CTYPE	Constraint type: either UPPER for upper bound or LOWER for lower bound (Text, Default = LOWER)
CLAREQ	Required flexible-to-rigid lift curve slope (Real ≠ 0.0)

Remarks:

1. Displacement constraints are selected in *Solution Control* with the discipline option:
 DCON=SID
2. A **LOWER** bound constraint excludes all values to the left of **CLAREQ** on a real number line, while an **UPPER** bound constraint excludes all values to the right, irrespective of the sign of **CLAREQ**.

Input Data Entry: DCONDSP**Description:** Defines a deflection constraint of the form:

$$\sum_j A_j u_j \leq \delta \text{ all (UPPER BOUND) or } \sum_j A_j u_j \geq \delta \text{ (LOWER BOUND)}$$

Format and Example:

1	2	3	4	5	6	7	8	9	10
DCONDSP	CTSET	DCID	CTYPE	DALL	LABEL	G	C	A	CONT
CONT		G	C	A	G	C	A	-etc-	

DCONDSP	1	10	LOWER	-2.3	TIP	32	3	2.0	ABC
+BC		7	3	-4.0					

Field	Contents
CTSET	Constraint set identification number (Integer > 0)
DCID	Constraint identification number (Integer > 0)
CTYPE	Constraint type, either UPPER or LOWER bound (Text, Default = UPPER)
DALL	Allowable displacement (Real)
LABEL	User specified label to identify constraint (Text)
G	Grid identification (Integer > 0)
C	Component number—any one of digits 1 through 6
A	Real coefficient (Real ≠ 0.0)

Remarks:

1. Displacement constraints are selected in Solution Control with the discipline option:

DCON=CTSET

The **CTSET** is the constraint set identification number and **DCID** is an arbitrary constraint identifier supplied by the user. All **DCONDSP** that share the same **CTSET** and **DCID** will form one constraint equation.

2. Both upper and lower bounds on the deflections can be specified by this entry. For example, if constraints of the form $|u| \leq 2.0$ are to be imposed, one **DCONDSP** entry would use **CTYPE** = **UPPER**, **DALL** = 2.0, G = 32, C = 3, A = 1.0 while a second entry would use **CTYPE** = **LOWER**, **DALL** = -2.0, G = 32, C = 3, A = 1.0.
3. Twist constraints can be specified by differencing two displacements while member constraints can be expressed as a weighted sum of three displacements.
4. Any number of continuation entries are permitted.
5. A **LOWER** bound constraint excludes all values to the left of **DALL** on a real number line, while an **UPPER** bound constraint excludes all values to the right, irrespective of the sign of **DALL**.

Input Data Entry DCONEP Principal Strain Constraint Definition

Description: Defines a principal strain constraint by specifying the identification numbers of constrained elements.

Format and Example:

1	2	3	4	5	6	7	8	9	10
DCONEP	SID	ST	SC	SS	ETYPE	LAYRNUM	EID1	EID2	CONT
CONT	EID3	EID4	-etc-						

DCONEP	100	1.-2	-1.-2	1.-2	BAR		101	102	ABC
+BC	107	108	142						

Alternate Form:

1	2	3	4	5	6	7	8	9	10
DCONEP	SID	ST	SC	SS	ETYPE	LAYRNUM	EID1	THRU	CONT
CONT	EID2								

Field	Contents
-------	----------

SID	Strain constraint set identification (Integer > 0)
ST	Principal strain limit in tension (Real > 0.0)
SS	Principal strain limit in compression (Real, Default = ST)
SS	Principal strain limit in shear (Real > 0.0)
ETYPE	Element type (Text)
LAYRNUM	Layer number of a composite element (Integer > 0 or Blank)
EID _i	Element identification numbers (Integer > 0)

Remarks:

- Strain constraints are selected in Solution Control with the discipline option:
STRAIN=sid
- ETYPE** may be selected from **BAR**, **QDMM1**, **QUAD4**, **ROD**, **SHEAR**, **TRIA3**, or **TRMEM**.
- If the alternate form is used, **EID2** must be greater than or equal to **EID1**. Elements in the range which do not exist are ignored.
- The shear strain limit, **SS**, is used only with the **SHEAR** element.
- The strain limit for compression, **SC**, is always treated as a negative value regardless of the sign of the input value.
- LAYRNUM** is only used if the element is composed of a composite material defined with **PCOMP** Bulk Data entries.

Input Data Entry **DCONEPM** Principal Strain Constraint Definition

Description: Defines a principal strain constraint by specifying material identification numbers.

Format and Example:

1	2	3	4	5	6	7	8	9	10
DCONEPM	SID	ST	SC	SS	MID1	MID2	MID3	MID4	CONT
CONT	MID5	MID6	-etc-						
DCONEPM	100	1.-2	-1.-2	1.-2	8888	9999	1	99	ABC
+BC	111	123							

Alternate Form:

1	2	3	4	5	6	7	8	9	10
DCONEPM	SID	ST	SC	SS	MAT1	THRU	MAT2		

Field

Contents

SID	Strain constraint set identification (Integer > 0)
ST	Principal strain limit in tension (Real > 0.0)
SC	Principal strain limit in compression (Real, Default = ST)
SS	Principal strain limit in shear (Real > 0.0)
MIDi	Material identification numbers (Integer > 0)

Remarks:

1. Strain constraints are selected in Solution Control with the discipline option:
STRAIN=sid
2. If the alternate form is used, **MID2** must be greater than or equal to **MID1**. Material properties in the range which do not exist are ignored.
3. The shear strain limit, **SS**, is used only with the **SHEAR** element.
4. The strain limit for compression, **SC**, is always treated as a negative value regardless of the sign of the input value.

Input Data Entry **DCONEPP** Principal Strain Constraint Definition

Description: Defines a principal strain constraint by specifying element property identification numbers.

Format and Example:

1	2	3	4	5	6	7	8	9	10
DCONEPP	SID	ST	SC	SS	PTYPE	LAYRNUM	PID1	PID2	CONT
CONT	PID3	PID4	-etc-						
DCONEPP	100	1.-2	-1.-2	1.-2	PBAR		100	200	ABC
+BC	300	400	500						

Alternate Form:

1	2	3	4	5	6	7	8	9	10
DCONEPP	SID	ST	SC	SS	PTYPE	LAYRNUM	PID1	THRU	ABC
+BC	PID2								

Field	Contents
SID	Strain constraint set identification (Integer > 0)
ST	Principal strain limit in tension (Real > 0.0)
SC	Principal strain limit in compression (Real, Default = ST)
SS	Principal strain limit in shear (Real > 0.0)
PTYPE	Property type (Text)
LAYRNUM	Layer number of a composite element (Integer > 0 or Blank)
PIDi	Property identification numbers (Integer > 0)

Remarks:

1. Strain constraints are selected in Solution Control with the discipline option:
STRAIN=sid
2. **PTYPE** may be selected from **PBAR**, **PCOMP**, **PCOMP1**, **PCOMP2**, **PQDMEM1**, **PROD**, **PSHEAR**, **PSHELL**, or **PTRMEM**.
3. If the alternate form is used, **PID2** must be greater than or equal to **PID1**. Property identification numbers in the range which do not exist are ignored.
4. The shear strain limit, **SS**, is used only with the **SHEAR** element.
5. The strain limit for compression, **SC**, is always treated as a negative value regardless of the sign of the input value.
6. **LAYRNUM** is only used if the element is composed of a composite material defined with **PCOMP** Bulk Data entries.

Input Data Entry: DCONFLT Flutter Constraint Definition

Description: Defines a flutter constraint in the form of a table:

$$\frac{Y - Y_{REQ}}{GFACT} \leq 0.0$$

Format and Example:

1	2	3	4	5	6	7	8	9	10
DCONFLT	SID	VTYPE	GFACT	V1	GAM1	V2	GAM2		CONT
CONT	V3	GAM3	V4	GAM4	-etc-				
DCONFLT	100	EQUIV	0.1	0.0	0.0	35.	0.05		

Field	Contents
SID	Constraint set identification, the constraints are referenced by the design constraint id in Solution Control (Integer > 0)
VTYPE	Nature of the velocity referred to in the table. Either TRUE for true velocity or EQUIV for equivalent air speed. Default = TRUE .
GFACT	Constraint scaling factor (Real > 0.0, Default = 0.10)
Vi	Velocity value (Real ≥ 0.0)
GAMi	Required damping value (Real)

Remarks:

- Flutter constraints are selected in Solution Control with the discipline option:
FLCOND=SID
- A negative value of **GAMi** refers to a stable system.
- The **Vi** must be in either ascending or descending order.
- Linear interpolation is used to determine **GAMA** for a given velocity.
- At least two pairs must be entered.
- Jumps between two points (**Vi = Vi+1**) are allowed, but not at the end points. If the jump point is used, the average of the two **GAMi** will be returned.

Input Data Entry: **DCONFRQ**

Description: Defines a frequency constraint of the form:

$$f \leq f_{all} \text{ or } f \geq f_{all}$$

Format and Example:

1	2	3	4	5	6	7	8	9	10
DCONFRQ	SID	MODE	CTYPE	FRQALL					
DCONFRQ	3	1	LOWER	6.0					

Field	Contents
SID	Constraint set identification (Integer > 0)
MODE	Modal number of the frequency to be constrained (Integer > 0)
CTYPE	Constraint type: either UPPER for upper bound or LOWER for lower bound (Text, Default = LOWER)
FRQALL	Frequency constraint (in Hz.).

Remarks:

1. More than one constraint can be placed on a mode allowing specification of pseudo-equality constraints.

Input Data Entry **DCONFT** Fiber/Transverse Strain Constraint Definition

Description: Defines fiber/transverse strain constraints for composite elements by specifying the identification numbers of constrained elements.

Format and Example:

1	2	3	4	5	6	7	8	9	10
DCONFT	SID	EFT	EFC	ETT	ETC	ETYPE	LAYRNUM	EID1	CONT
CONT	EID2	EID3	-etc-						

DCONFT	100	1.-2	-1.-2	1.-3	-1.-3	QUAD4	1	101	ABC
+BC	102	110							

Alternate Form:

1	2	3	4	5	6	7	8	9	10
DCONFT	SID	EFT	EFC	ETT	ETC	ETYPE	LAYRNUM	EID1	CONT
CONT	THRU	EID2							

Field	Contents
SID	Strain constraint set identification (Integer > 0)
EFT	Tensile strain limit in the fiber direction (Real > 0.0)
EFC	Compressive strain limit in the fiber direction (Real, Default = EFT)
ETT	Tensile strain limit in the transverse direction (Real > 0.0)
ETC	Compressive strain limit in the transverse direction (Real, Default = ETT)
ETYPE	Element type (Text)
LAYRNUM	The layer number of a composite element (Integer > 0, or blank)
EIDi	Element identification numbers (Integer > 0)

Remarks:

- Strain constraints are selected in Solution Control with the discipline option:
STRAIN=sid
- Fiber/transverse strain constraints may only be applied to elements defined using composite materials.
- ETYPE** may be selected from **QDNEM1**, **QUAD4**, **TRIA3**, or **TRMEM**.
- If the alternate form is used, **EID2** must be greater than or equal to **EID1**. Elements in the range which do not exist are ignored.
- The strain limits for compression, **EFC** and **ETC**, are always treated as negative values regardless of the signs of the input values.

Input Data Entry DCONFTM Fiber/Transverse Strain Constraint Definition

Description: Defines fiber/transverse strain constraints for composite elements by specifying material identification numbers.

Format and Example:

1	2	3	4	5	6	7	8	9	10
DCONFTM	SID	EFT	EFC	ETT	ETC	MID1	MID2	MID3	CONT
CONT	MID4	MID5	-etc-						

DCONFTM	100	1.-2	-1.-2	1.-3	-1.-3	11	16	101	ABC
+BC	19	14							

Alternate Form:

1	2	3	4	5	6	7	8	9	10
DCONFTM	SID	EFT	EFC	ETT	ETC	MID1	THRU	MID2	

Field	Contents
SID	Strain constraint set identification (Integer > 0)
EFT	Tensile strain limit in the fiber direction (Real > 0.0).
EFC	Compressive strain limit in the fiber direction (Real, Default = EFT)
ETT	Tensile strain limit in the transverse direction (Real > 0.0)
ETC	Compressive strain limit in the transverse direction (Real, Default = ETT).
MID _i	Material identification numbers (Integer > 0)

Remarks:

1. Strain constraints are selected in Solution Control with the discipline option:
STRAIN=sid
2. Fiber/transverse strain constraints may only be applied to elements defined using composite materials.
3. If the alternate form is used, **MID2** must be greater than or equal to **MID1**. Material properties in the range which do not exist are ignored.
4. The strain limits for compression, **EFC** and **ETC**, are always treated as negative values regardless of the signs of the input values.

Input Data Entry **DCONFTP** Fiber/Transverse Strain Constraint Definition

Description: Defines fiber/transverse strain constraints for composite elements by specifying property identification numbers.

Format and Example:

1	2	3	4	5	6	7	8	9	10
DCONFTP	SID	EFT	EFC	ETT	ETC	PTYPE	LAYRNUM	PID1	CONT
CONT	PID2	PID3	-etc-						
DCONFTP	100	1.-2	1.-2	2.-3	3.-3	PCOMP	2	100	CONT
CONT	110	120							

Alternate Form:

1	2	3	4	5		7	8	9	10
DCONFTP	SID	EFT	EFC	ETT	ETC	ETYPE	LAYRNUM	PID1	CONT
CONT	THRU	PID2							

Field	Contents
SID	Strain constraint set identification (Integer > 0).
EFT	Tensile strain limit in the fiber direction (Real > 0.0)
EFC	Compressive strain limit in the fiber direction (Real, Default = EFT)
ETT	Tensile strain limit in the transverse direction (Real > 0.0).
ETC	Compressive strain limit in the transverse direction (Real, Default = ETT)
PTYPE	Property type (Text)
LAYRNUM	The layer number of a composite element (Integer > 0 or blank)
PIDi	Property identification numbers (Integer > 0)

Remarks:

1. Strain constraints are selected in Solution Control with the discipline option:
STRAIN=sid
2. **PTYPE** may be selected from **PCOMP**, **PCOMP1**, and **PCOMP2**.
3. Fiber/transverse strain constraints may only be applied to elements defined using composite materials.
4. If the alternate form is used, **PID2** must be greater than or equal to **PID1**. Properties in the range which do not exist are ignored.
5. The strain limits for compression, **EFC** and **ETC**, are always treated as negative values regardless of the signs of the input values.

Input Data Entry: **DCONLAM** Composite laminate composition constraint.

Description: Defines a constraint on the relative thickness of a *ply* that is part of a *laminate*. The constraint is of the form:

$$\frac{\%req}{100} - \frac{t_{ply}}{t_{lam}} \leq 0 \quad (\text{lower bound})$$

$$\frac{t_{ply}}{t_{lam}} - \frac{\%req}{100} \leq 0 \quad (\text{upper bound})$$

Format and Example:

1	2	3	4	5	6	7	8	9	10
DCONLAM	CTYPE	%REQ	PLYNUM	PLYSET	LAM	SID	SID	SID	CONT
CONT	SID	SID	-etc-						
DCONLAM	UPPER	40.0		100	ALL	1000	1001		

Field	Contents
CTYPE	Constraint type: either UPPER for upper bound or LOWER for lower bound. (Text, Default = UPPER)
%REQ	Minimum (lower bound) or maximum (upper bound) PERCENTAGE (0.0 to 100.0) of the total laminate thickness that is to be made up of the ply thickness. (see Remark 2) (Real > 0.0)
PLYNUM	Single ply number (numbered in the order used on the PCOMPi) that constitutes the ply thickness. Only one of PLYNUM or PLYSET may be used. (Integer > 0 or blank)
PLYSET	Set identification number of one or more PLYLIST bulk data entries naming a set of plies whose summed thicknesses constitute the <i>ply</i> thickness in the constraint. Only one of PLYNUM or PLYSET may be used. (Integer > 0 or blank)
LAM	The character string ALL or the set identification number of one or more PLYLIST entries naming a set of plies whose summed thicknesses constitute the <i>laminate</i> thickness in the constraint. If ALL , the <i>laminate</i> is defined to be all the layers on the PCOMPs of the elements selected by SIDi . (Character = ALL or Integer > 0, Default = ALL)
SID	Set identification of one or more ELEMLIST entries that define the set of composite elements to which this composition constraint will be applied. (Integer > 0 or blank)

Remarks:

1. One and only one of either **PLYNUM** or **PLYSET** must be given.
2. The definition of *ply* and *laminate* thickness can vary from entry to entry. **If** **PLYNUM** is used to define t_{ply} that one layer constitutes a ply; otherwise t_{ply} is the **sum** of the layer thicknesses of all the layers listed in **PLYSET**.

Similar rules are applied for *tlam*. If **ALL** is used, every layer of the element is used to compute *tlam* (including undesigned layers-see Remark 3); otherwise the *summed* thicknesses of the layers specified by the **PLYLIST** set will be used. As a result, there is no real distinction between a *ply* thickness and a *laminate* thickness. Typically, the *ply* will be a subset of the layers that define the *laminate*, but that is not a requirement.

3. If this constraint is applied to a composite element with undesigned layers, these layers may be freely included in the layer(s) composing the *ply* and/or the layer(s) composing the *laminate*. The only restriction is that at least one layer in the *ply* must be a local design variable **and** at least one layer in the *laminate* must be a local design variable.

Input Data Entry: **DCONLMN** Composite laminate minimum gauge constraint.

Description: Defines a lower bound constraint on the total thickness of all or part of the layers of a composite element. The constraint is of the form:

$$1.0 - \frac{t_{lam}}{t_{min}} \leq 0$$

Format and Example:

1	2	3	4	5	6	7	8	9	10
DCONLMN	MINTHK	LAM	SID	SID	SID	SID	SID	SID	CONT
CONT	SID	SID	-etc-						
DCONLMN	0.20	ALL	1001	1002				,	

Field	Contents
MINTHK	Minimum <i>laminate</i> thickness. (Real > 0.0, Default = 10 ⁻⁴)
LAM	The character string ALL or the set identification number of one or more PLYLIST entries naming a set of plies whose <i>summed</i> thicknesses constitute the <i>laminate</i> thickness in the constraint. If ALL , the <i>laminate</i> is defined to be all the layers on the PCOMPs of the elements selected by SIDi . (Character = ALL or Integer > 0, Default = ALL)
SID	Set identification of one or more ELEMLIST entries that define the set of composite elements to which this composition constraint will be applied. (Integer > 0 or blank)

Remarks:

1. Because of the generality of the definition of the *laminate*, there is no real distinction between the **DCONLMN** and the **DCONPMN** constraints. Only the defaults are different to allow simple definitions of the common laminate in **DCONLMN** (**ALL**) or ply (**PLYNUM**) in **DCONPMN**.
2. The definition of *laminate* thickness can vary from entry to entry. If **ALL** is used, every layer of the element is used to compute *t_{lam}* (including undesigned layers-see Remark 3); otherwise the *summed* thicknesses of the layers specified by the **PLYLIST** set will be used.
3. If this constraint is applied to a composite element with undesigned layers, these layers may be freely included in the layer(s) composing the *ply* and/or the layer(s) composing the *laminate*. The only restriction is that at least one layer in the *laminate* must be a local design variable.
4. If the *laminate* is composed of a single layer, this constraint becomes redundant with the **TMIN** entered on the **PCOMP_i** field (for shape function linking) or the **VMIN** entered on the **DESELM** or **DESVARP** entry (for physical linking). In this case, the most critical limit will be determined from among all sources (**DCONPMN**, **DCONLMN**, **TMIN/VMIN**) and will be used to update the local variable side constraint. The **DCONxxx** entry will then be automatically removed since it will no longer be necessary. A summary of this action will be echoed to the print file.

Input Data Entry: **DCONPMN** Composite element ply minimum gauge constraint.

Description: Defines a lower bound constraint on the total thickness of all or part of the layers of a composite element. The constraint is of the form:

$$1.0 - \frac{t_{ply}}{t_{min}} \leq 0$$

Format and Example:

1	2	3	4	5	6	7	8	9	10
DCONPMN	MINTHK	PLYNUM	PLYSET	SID	SID	SID	SID	SID	CONT
CONT	SID	SID	-etc-						
DCONPMN	0.010	3		1001	1002			,	

Field	Contents
MINTHK	Minimum <i>ply</i> thickness. (Real > 0.0, Default = 10^{-4})
PLYNUM	Single <i>ply</i> number (numbered in the order used on the PCOMP1) that constitutes the <i>ply</i> thickness. Only one of PLYNUM or PLYSET may be used. (Integer > 0 or blank)
PLYSET	Set identification number of one or more PLYLIST bulk data entries naming a set of plies whose <i>summed</i> thicknesses constitute the <i>ply</i> thickness in the constraint. Only one of PLYNUM or PLYSET may be used. (Integer > 0 or blank)
SID	Set identification of one or more ELEMLIST entries that define the set of composite elements to which this composition constraint will be applied. (Integer > 0 or blank)

Remarks:

1. One and only one of either **PLYNUM** or **PLYSET** must be given.
2. Because of the generality of the definition of the *ply*, there is no real distinction between the **DCONLMN** and the **DCONPMN** constraints. Only the defaults are different to allow simple definitions of the common laminate in **DCONLMN** (**ALL**) or *ply* (**PLYNUM**) in **DCONPMN**.
3. The definition of *ply* thickness can vary from entry to entry. If **PLYNUM** is used to define t_{ply} , that one layer constitutes a *ply*; otherwise t_{ply} is the *sum* of the layer thicknesses of all the layers listed in **PLYSET**.
4. If this constraint is applied to a composite element with undesigned layers, these layers may be freely included in the layer(s) composing the *ply*. The only restriction is that at least one layer in the *ply* must be a local design variable.
5. If the *ply* is composed of a single layer, this constraint becomes redundant with the **TMIN** entered on the **PCOMP1** field (for shape function linking) or the **VMIN** entered on the **DESELM** or **DESVARP** entry (for physical linking). In this case, the most critical limit will be determined from among all sources (**DCONPMN**, **DCONLMN**, **TMIN/VMIN**) and will be used to update the local variable side constraint. The **DCONxxx** entry will then be automatically removed since it will no longer be necessary. A summary of this action will be echoed to the print file.

Input Data Entry **DCONLIST** Design Constraint List

Description: Defines a list of design constraints for which constraint value output and/or constraint gradient output are desired.

Format and Example:

1	2	3	4	5	6	7	8	9	10
DCONLIST	SID	TYPE	NRFAC	EPS					
DCONLIST	1000	DISP	0.6	-.05					

Field	Contents
SID	Set identification number (Integer > 0)
TYPE	The design constraint type. One of the following: FREQ frequency FLUT flutter DISP displacement VMISES Von Mises TSAIWU Tsai-Wu STRAIN strain THICK thickness EFF aeroelastic effectiveness SCF stability coefficient TRIM trim ALL all of the above OTHER all EXCEPT the above The Default value is ALL
NRFAC	Constraint retention factor for math programming methods. At least NRFAC * (number of design variables) constraints will be considered active. (Real > 0.0, Default = 3.0)
EPS	Constraint retention parameter in which all constraints having a value greater than EPS will be considered active. (Real, Default = - 0.1)

Remarks:

1. **NRFAC** and **EPS** control the number of constraints that are selected for print and punch output. For constraint gradients, only those considered active by the global constraint screening algorithm (**NRFAC** and **EPS** from the **OPTIMIZE** command in Solution control) are available to be selected.
2. More than one **DCONLIST** with the same set identification number may be used to select subsets of different constraint types.

Input Data Entry DCONSCF Stability Derivative Constraint

Description: Defines a constraint on the flexible stability derivative at the reference grid point associated with the force or moment due to a trim parameter or control surface deflection of the form:

$$\left[\frac{\partial C_F}{\partial \delta_{trim}} \right]_{lower} \leq \frac{\partial C_F}{\partial \delta_{trim}} \leq \left[\frac{\partial C_F}{\partial \delta_{trim}} \right]_{upper}$$

Format and Example:

1	2	3	4	5	6	7	8	9	10
DCONSCF	SETID	ACCLAB	PRMLAB	CTYPE	PRMREQ	UNITS			
DCONSCF	999	PACCEL	AILERON	LOWER	1.0	RADIANS			

Field	Contents
SETID	Set identification number referenced by the DCONSTRAINT Solution Control option of the SAERO command. (Integer > 0)
ACCLAB	Alphanumeric string identifying the aerodynamic force or moment by naming the corresponding structural acceleration in a manner consistent with the TRIM entry. See Remarks 2 and 4.
PRMLAB	Alphanumeric string identifying a constrained control surface or aeroelastic trim parameter (e.g. ALPHA or PRATE). See Remarks 3 and 4.
CTYPE	Constraint type; either UPPER , for upper bound, or LOWER for lower bound. (Character, default= UPPER)
PRMREQ	Bound for the stability coefficient. For units, see Remarks 5 and 6. (Real)
UNITS	Units for the stability coefficient. Either RADIANS or DEGREES . See Remark 6. (Real, Default= DEGREES)

Remarks:

1. The **DCONSCF** entry is selected in Solution Control with the **DCONSTRAINT=SETID** option of the **SAERO** command.
2. The **ACCLAB** may refer to any of the **TRIM** Bulk Data entry trim parameters that are **structural accelerations**. Valid trim parameters are **NX**, **NY**, **NZ**, **PACCEL**, **QACCEL**, and **RACCEL**.
3. The **PRMLAB** may refer to **AESURF** or **CONLINK** control surfaces or to any of the **TRIM** entry parameters **except the structural accelerations**. Valid selections are: **PRATE**, **QRATE**, **RRATE**, **ALPHA**, **BETA**, **THRCAM** and any control surface label. Invalid trim parameters are: **NX**, **NY**, **NZ**, **PACCEL**, **QACCEL** and **RACCEL**.
4. Any combination of forces or moments and trim parameters/control surfaces may be used on this entry **provided** they have the same symmetry as the associated **TRIM** entry. Furthermore, to apply the constraint to the flexible derivative, the degree of freedom corresponding to the force or moment must be supported in the boundary condition. For example, to constrain the pitching moment, **QACCEL**, due to angle of attack, **ALPHA**, the y-rotation of the support point must be on the **SUPPORT** entry for the boundary condition in which the **TRIM** is analyzed.

5. The stability derivatives are nondimensional quantities derived from the flexible forces and moments due to "unit" parameters. The constraint is applied to the nondimensional derivative at the user-defined reference point. To assist the defining **PRMREQ**, the following normalizations are used in **ASTROS**:

SYMMETRIC DERIVATIVES	FORCES	CONTROL SURFACES	stability coeff = $F/(QDP*S)$
		RATES	stability coeff = $F*2*VO/(QDP*S*C)$ "unit" rate = unit dimensional rate * $C/2*VO$
	MOMENTS	CONTROL SURFACES	stability coeff = $M/(QDP*S*C)$
		RATES	stability coeff = $M*4*VO/(QDP*S*C**2)$ "unit" rate = unit dimensional rate * $C/2*VO$
ANTISYMMETRIC DERIVATIVES	FORCES	CONTROL SURFACES	stability coeff = $F/(QDP*S)$
		RATES	stability coeff = $F*2*VO/(QDP*S*B)$ "unit" rate = unit dimensional rate * $C/2*VO$
	MOMENTS	CONTROL SURFACES	stability coeff = $M/(QDP*S*B)$
		RATES	stability coeff = $M*4*VO/(QDP*S*B**2)$ "unit" rate = unit dimensional rate * $B/2*VO$

F and **M** are the dimensional flexible forces and moments for the full vehicle; **S**, **C**, and **B** are the non-dimensional factors from the **AEROS** Bulk Data entry (the inputs are assumed to be for the full vehicle); and **QDP** and **VO** are defined on the **TRIM** Bulk Data entry.

6. **RADIANS** or **DEGREES** refer to the units of the unit control surface deflection or unit rate. **RADIANS** imply the value due to a unit RAD or RAD/S while **DEGREES** imply the value due to a unit DEG or DEG/S. **THRCAM** has no valid angular unit, hence the **UNITS** field is ignored.
7. A **LOWER** bound constraint excludes all values to the left of **PRMREQ** on a real number line, while an **UPPER** bound excludes all values to the right, irrespective of the sign of **PRMREQ**.

Input Data Entry: **DCONTH2** Thickness constraints on layers of composite elements

Description: Defines a set of layers for a list of elements linked using **SHAPE** entries for which the ply thickness constraints are to be retained on all design iterations.

Format and Example:

1	2	3	4	5	6	7	8	9	10
DCONTH2	ETYPE	LAYRNUM	LAYRLST	EID	EID	EID	EID	EID	CONT
CONT	EID	EID	-etc-						

DCONTHK	QUAD4		100	101	200	205			
---------	-------	--	-----	-----	-----	-----	--	--	--

Alternate Form:

1	2	3	4	5	6	7	8	9	10
DCONTH2	ETYPE	LAYRNUM	LAYRLST	EID	THRU	EID			

Field	Contents
ETYPE	Character input identifying the element type. One of the following: <div style="display: flex; justify-content: space-around; margin-top: 5px;"> QUAD4 TRIA3 </div> <div style="display: flex; justify-content: space-around; margin-top: 5px;"> QDMEM1 TRMEM </div>
LAYRNUM	Layer number of the layer(s) to be retained. The given layer will be retained for each element in the list of elements (Integer > 0 or blank, See Remark 1)
LAYRLST	Set identification number of a PLYLIST bulk data entry naming a set of plies to be retained as active for each element. (Integer > 0 or blank, See Remark 1)
EID	Element identification number (Integer > 0 or blank)

Remarks:

- One and only one of either **LAYRNUM** or **LAYRLST** must be given. Noncomposite elements must be called out on **DCONTHK** entries.
- The purpose of this bulk data list is to ensure that adequate physical move limits are retained in optimization with shape function design variable linking without requiring retention of all move limits. For problems with large numbers of local variables using shape functions, the move limits often cause too many minimum thickness constraints (see Remark 3) to be retained in the optimization task. Using this bulk data entry *or* its noncomposite counterpart **DCONTHK** to name "critical" minimum gauge constraints (see Remark 4) will cause only the named elements' thickness constraints to be computed and retained. All layers of composite elements named on **DCONTHK** will be retained.
NOTE that all elements' thickness constraints will always be computed irrespective of the **DCONTHK** entries, but may be deleted in the constraint deletion.
- The global design variable in shape function linking is non-physical and no reasonable restriction for a global variable move limit (side constraint) can be defined. Therefore, constraints on the local design variables controlled by shape functions are generated by ASTROS to ensure that the design is reasonable (ie, nonnegative thicknesses).

4. The **DCONTH2** entry should select a minimum number of elements linked to shape functions that will enable the optimizer to select physically reasonable designs without retaining all the minimum thickness constraints (potentially a very large number). Typically, this means $N+1$ elements spread over the range of the shape function (e.g. span or chord) where N is the order of the shape (**N=0**, **UNIFORM**; **N=1**, **LINEAR**, etc.).

Input Data Entry: **DCONTHK** Thickness constraints on elements

Description: Defines a list of elements linked using **SHAPE** entries for which thickness constraints are to be retained on all design iterations.

Format and Example:

1	2	3	4	5	6	7	8	9	10
DCONTHK	ETYPE	EID	EID	EID	EID	EID	EID	EID	CONT
CONT	EID	EID	-etc-						

DCONTHK	QDMEM1	100	101	200	205				
---------	--------	-----	-----	-----	-----	--	--	--	--

Alternate Form:

1	2	3	4	5	6	7	8	9	10
DCONTHK	ETYPE	EID	THRU	EID					

Field	Contents
ETYPE	Character input identifying the element type. One of the following:
	BAR QUAD4
	CONM2 ROD
	ELAS SHEAR
	MASS TRIA3
	QDMEM1 TRMEM
EID	Element identification number (Integer > 0 or blank)

Field

Contents

ETYPE Character input identifying the element type. One of the following:

BAR	QUAD4
CONM2	ROD
ELAS	SHEAR
MASS	TRIA3
QDMEM1	TRMEM

EID Element identification number (Integer > 0 or blank)

Remarks:

1. The purpose of this bulk data list is to ensure that adequate physical move limits are retained in optimization with shape function design variable linking without requiring retention of all move limits. For problems with large numbers of local variables using shape functions, the move limits often cause too many minimum thickness constraints (see Remark 2) to be retained in the optimization task. Using this bulk data entry OR its composite counterpart **DCONTH2** to name "critical" minimum gauge constraints (see Remark 3) will cause only the named elements' thickness constraints to be computed and retained. All layers of composite elements named on **DCONTHK** will be retained.
NOTE that all elements' thickness constraints will always be computed irrespective of the **DCONTHK** entries, but may be deleted in the constraint deletion.
2. The global design variable in shape function linking is non-physical and no reasonable restriction for a global variable move limit (side constraint) can be defined. Therefore, constraints on the local design variables controlled by shape functions are generated by ASTROS to ensure that the design is reasonable (ie, nonnegative thicknesses).
3. The **DCONTHK** entry should select a minimum number of elements linked to shape functions that will enable the optimizer to select physically reasonable designs without retaining all the minimum thickness constraints (potentially a very large number). Typically, this means N+1 elements spread over the range of the shape function (e.g. span or chord) where N is the order of the shape (N=0, UNIFORM; N=1, LINEAR, etc.). Use **DCONTH2** for composite elements in which linking across layers may allow certain layers to be omitted from the retention set.

Input Data Entry **DCONTRM** Aeroelastic Trim Parameter Constraint

Description: Defines a trim parameter constraint of the form:

$$\delta_{trim} \leq \delta_{trimReq} \quad \text{or} \quad \delta_{trim} \geq \delta_{trimReq}$$

Format and Example:

1	2	3	4	5	6	7	8	9	10
DCONTRM	SETID	PRMLAB	CTYPE	PRMREQ					
DCONTRM	100	AILERON	UPPER	25.0					

Field	Contents
SETID	Set identification number referenced by the DCONSTRAINT Solution Control command. (Integer > 0)
PRMLAB	Alphanumeric string identifying a constrained control surface or aeroelastic trim parameter (e.g. ALPHA or PRATE). (See Remark 2.)
CTYPE	Constraint type; either UPPER , for upper bound, or LOWER for lower bound. (Character, Default = UPPER)
PRMREQ	Bound for the trim parameter. For units, see Remark 3. (Real)

Remarks:

1. The **DCONTRM** entry is selected in Solution Control with the **DCONSTRAINT=SETID** option of the **SAERO** command.
2. The **PRMLAB** may refer to **AESURF** or **CONLINK** control surfaces or to any of the **TRIM** entry parameters, **NX**, **NY**, **NZ**, **PACCEL**, **QACCEL**, **RACCEL**, **PRATE**, **QRATE**, **RRATE**, **ALPHA**, or **BETA**. The only requirement is that the constrained control surface must be declared on the **TRIM** entry. The user will be **warned** if trim parameters not on the **TRIM** entry are constrained (since these parameters are fixed, they are design invariant).
3. The units for control surface deflections are degrees. For rates, the units should be radians/sec. For linear accelerations **NX**, **NY**, **NZ**, the units should be consistent, (length/sec/sec) or, if a **CONVERT,MASS** entry was used, should be dimensionless. Angular accelerations should be in radians/sec/sec.
4. A **LOWER** bound constraint excludes all values to the left of **PRMREQ** on a real number line, while an **UPPER** bound excludes all values to the right, irrespective of the sign of **PRMREQ**.

Input Data Entry DCONTW Tsai-Wu Stress Constraint Definition

Description: Defines Tsai-Wu stress constraints by specifying the identification numbers of constrained elements

Format and Example:

1	2	3	4	5	6	7	8	9	10
DCONTW	SID	XT	XC	YT	YC	SS	F12	ETYPE	CONT
CONT	LAYRNUM	EID1	EID2	EID3	-etc-				

DCONTW	100	1.+6	-1.+6	1.+4	-1.+4	1.5+3		QUAD4	ABC
+BC	1	102	106	110					

Alternate Form:

1	2	3	4	5	6	7	8	9	10
DCONTW	SID	XT	XC	YT	YC	SS	F12	ETYPE	CONT
CONT	LAYRNUM	EID1	THRU	EID2					

Field	Contents
SID	Stress constraint set identification (Integer > 0)
XT	Tensile stress limit in the longitudinal direction (Real > 0.0)
XC	Compressive stress limit in the longitudinal direction (Real, Default = XT)
YT	Tensile stress limit in the transverse direction (Real > 0.0)
YC	Compressive stress limit in the transverse direction (Real, Default = YT)
SS	Shear stress limit for in-plane stress (Real > 0.0)
F12	Tsai-Wu interaction term (Real)
ETYPE	Element type (Text)
LAYRNUM	The layer number of a composite element (Integer > 0 or blank)
EID _i	Element identification numbers (Integer > 0)

Remarks:

1. Stress constraints are selected in Solution Control with the discipline option:
STRESS=sid
2. **ETYPE** may be selected from **QDMEM1**, **QUAD4**, **TRIA3**, or **TRMEM**.
3. If the alternate form is used, **EID2** must be greater than or equal to **EID1**. Elements in the range which do not exist are ignored.
4. The strain limits for compression, **XC** and **YC**, are always treated as negative values regardless of the sign of the input values.
5. **LAYRNUM** is only used if the element is composed of a composite material defined with **PCOMP** Bulk Data entries.

Input Data Entry DCONTWM Tsai-Wu Stress Constraint Definition

Description: Defines Tsai-Wu stress constraints by specifying material identification numbers

Format and Example:

1	2	3	4	5	6	7	8	9	10
DCONTWM	SID	XT	XC	YT	YC	SS	F12	MID1	CONT
CONT	MID2	MID3	MID4	-etc-					

DCONTWM	100	1.+6	-1.+6	1.+4	-1.+4	1.5+3		101	ABC
+BC	102	200	310						

Alternate Form:

1	2	3	4	5	6	7	8	9	10
DCONTWM	SID	XT	XC	YT	YC	SS	F12	MID1	CONT
CONT	THRU	MID2							

Field**Contents**

SID	Stress constraint set identification (Integer > 0)
XT	Tensile stress limit in the longitudinal direction (Real > 0.0)
XC	Compressive stress limit in the longitudinal direction (Real, Default = XT)
YT	Tensile stress limit in the transverse direction (Real > 0.0)
YC	Compressive stress limit in the transverse direction (Real, Default = YT)
SS	Shear stress limit for in-plane stress (Real > 0.0)
F12	Tsai-Wu interaction term (Real)
MID1	Material identification numbers (Integer > 0)

Remarks:

1. Stress constraints are selected in Solution Control with the discipline option:
STRESS=sid
2. If the alternate form is used, **MID2** must be greater than or equal to **MID1**. Materials in the range which do not exist are ignored.
3. The stress limits for compression, **XC** and **YC**, are always treated as negative values regardless of the sign of the input values.

Input Data Entry **DCONTWP** Tsai-Wu Stress Constraint Definition

Description: Defines Tsai-Wu stress constraints by specifying element property identification numbers

Format and Example:

1	2	3	4	5	6	7	8	9	10
DCONTWP	SID	XT	XC	YT	YC	SS	F12	PTYPE	CONT
CONT	LAYRNUM	PID1	PID2	PID3	-etc-				

DCONTWP	100	1.+6	-1.+6	1.+4	-1.+4	1.5+3		PCOMP	ABC
+BC	100	200	300						

Alternate Form:

1	2	3	4	5	6	7	8	9	10
DCONTWP	SID	XT	XC	YT	YC	SS	F12	PTYPE	CONT
CONT	LAYRNUM	PID1	THRU	PID2					

Field	Contents
SID	Stress constraint set identification (Integer > 0)
XT	Tensile stress limit in the longitudinal direction (Real > 0.0)
XC	Compressive stress limit in the longitudinal direction (Real, Default = XT)
YT	Tensile stress limit in the transverse direction (Real > 0.0)
YC	Compressive stress limit in the transverse direction (Real, Default = YT)
SS	Shear stress limit for in-plane stress (Real > 0.0)
F12	Tsai-Wu interaction term (Real)
PTYPE	Property type (Text)
LAYRNUM	The layer number of a composite element (Integer > 0 or blank)
PIDi	Property identification numbers (Integer > 0)

Remarks:

1. Stress constraints are selected in Solution Control with the discipline option:
STRESS=sid
2. **PTYPE** may be selected from **PCOMP**, **PCOMP1**, **PCOMP2**, **PQDMEM1**, **PSHELL**, or **PTRMEM**.
3. If the alternate form is used, **PID2** must be greater than or equal to **PID1**. Properties in the range which do not exist are ignored.
4. The stress limits for compression, **XC** and **YC**, are always treated as negative values regardless of the sign of the input values.
5. **LAYRNUM** is only used if the element is composed of a composite material defined with **PCOMP** Bulk Data entries.

Input Data Entry DCONVM Von-Mises Stress Constraint Definition

Description: Defines a Von-Mises stress constraint by specifying the identification numbers of constrained elements

Format and Example:

1	2	3	4	5	6	7	8	9	10
DCONVM	SID	ST	SC	SS	ETYPE	LAYRNUM	EID1	EID2	CONT
CONT	EID3	EID4	-etc-						

DCONVM	100	1.+6	-1.+6	1.+4	BAR		101	102	ABC
+BC	107	108	142						

Alternate Form:

1	2	3	4	5	6	7	8	9	10
DCONVM	SID	ST	SC	SS	ETYPE	LAYRNUM	EID1	THRU	CONT
CONT	EID2								

Field	Contents
SID	Stress constraint set identification (Integer > 0)
ST	Tensile stress limit (Real > 0.0 or blank)
SC	Compressive stress limit (Real, Default = ST).
SS	Shear stress limit (Real > 0.0 or blank)
ETYPE	Element type (Text)
LAYRNUM	The layer number of a composite element (Integer > 0 or blank)
EID _i	Element identification numbers (Integer > 0)

Remarks:

- Stress constraints are selected in Solution Control with the discipline option:
STRESS=aid
- ETYPE** may be selected from **BAR**, **QDMM1**, **QUAD4**, **ROD**, **SHEAR**, **TRIA3**, or **TRMEM**.
- If the alternate form is used, **EID2** must be greater than or equal to **EID1**. Elements in the range which do not exist are ignored.
- The stress limit for compression, **SC**, is always treated as a negative value regardless of the sign of the input value.
- LAYRNUM** is only used if the element is composed of a composite material defined with **PCOMP** Bulk Data entries.

Input Data Entry DCONVMM Von-Mises Stress Constraint Definition

Description: Defines a Von-Mises stress constraint by specifying material identification numbers.

Format and Example:

1	2	3	4	5	6	7	8	9	10
DCONVMM	SID	ST	SC	SS	MID1	MID2	MID3	MID4	CONT
CONT	MID5	MID6	-etc-						

DCONVMM	100	1.+6	-1.+6	1.+4	101	201	301	401	ABC
+BC	501	601	701						

Alternate Form:

1	2	3	4	5	6	7	8	9	10
DCONVMM	SID	ST	SC	SS	MID1	THRU	MID2		

Field	Contents
SID	Stress constraint set identification (Integer > 0)
ST	Tensile stress limit (Real > 0.0 or blank)
SC	Compressive stress limit (Real, Default = ST)
SS	Shear stress limit (Real > 0.0 or blank)
MIDi	Material identification numbers (Integer > 0)

Remarks:

1. Stress constraints are selected in Solution Control with the discipline option:
STRESS=sid
2. If the alternate form is used, **MID2** must be greater than or equal to **MID1**. Materials in the range which do not exist are ignored.
3. The stress limit for compression, **SC**, is always treated as a negative value regardless of the sign of the input value.

Input Data Entry DCONVMP Von-Mises Stress Constraint Definition

Description: Defines a Von-Mises stress constraint by specifying property identification numbers.

Format and Example:

1	2	3	4	5	6	7	8	9	10
DCONVMP	SID	ST	SC	SS	PTYPE	LAYRNUM	PID1	PID2	CONT
CONT	PID3	PID4	-etc-						

DCONVMP	100	1.+6	-1.+6	1.+4	PBAR		102	103	ABC
+BC	107	108	142						

Alternate Form:

1	2	3	4	5	6	7	8	9	10
DCONVMP	SID	ST	SC	SS	PTYPE	LAYRNUM	PID1	THRU	CONT
CONT	PID2								

Field	Contents
SID	Stress constraint set identification (Integer > 0).
ST	Tensile stress limit (Real > 0.0 or blank)
SC	Compressive stress limit (Real, Default = ST)
SS	Shear stress limit (Real > 0.0 or blank)
PTYPE	Property type (Text)
LAYRNUM	The layer number of a composite element (Integer > 0 or blank)
PIDi	Property identification numbers (Integer > 0)

Remarks:

1. Stress constraints are selected in Solution Control with the discipline option:
STRESS=sid
2. **PTYPE** may be selected from **PBAR**, **PCOMP**, **PCOMP1**, **PCOMP2**, **PQDMEM1**, **PROD**, **PSHEAR**, **PSHELL**, or **PTRMEM**.
3. If the alternate form is used, **PID2** must be greater than or equal to **PID1**. Properties in the range which do not exist are ignored.
4. The stress limit for compression, **SC**, is always treated as a negative value regardless of the sign of the input value.
5. **LAYRNUM** is only used if the element is composed of a composite material defined with **PCOMP** Bulk Data entries.

Input Data Entry: DESELM

Description: Designates design variable properties when the design variable is uniquely associated with a single finite element

Format and Example:

1	2	3	4	5	6	7	8	9	10
DESELM	DVID	EID	ETYPE	VMIN	VMAX	VINIT	LAYERNUM	LABEL	
DESELM	5	10	CROD	0.01	10.0	1.0			

Field	Contents
DVID	Design variable identification (Integer > 0)
EID	Element identification (Integer > 0)
ETYPE	Element type
VMIN	Minimum allowable value of the design variable (Real ≥ 0.0) (Default = .001)
VMAX	Maximum allowable value of the design variable (Real ≥ 0.0) (Default = 1000.)
VINIT	Initial value of the design variable (Real, $VMIN \leq VINIT \leq VMAX$) (Default is $VMIN$)
LAYERNUM	The layer number of a composite element to be designed (Integer > 0, or blank)
LABEL	Optional user-supplied label to define the design variable (text)

Remarks:

1. Valid **ETYPE**s are **CROD**, **CONROD**, **CBAR**, **CSHEAR**, **CTRIA3**, **CTRMEM**, **CQDNEM1**, **CQUAD4**, **CELAS1**, **CELAS2**, **CMASS1**, **CMASS2** and **CONM2**.
2. The initial element thickness or area used in the structural analysis is derived from the **VINIT** value and the property value on the associated property entry:

$$t_{init} = VINIT * property_value$$

Similarly, the minimum and maximum values are the **VMIN** and **VMAX** values of the element property are derived from:

$$t_{min} = VMIN * property_value$$

$$t_{max} = VMAX * property_value$$

3. **DVID** must be unique among all **DESELM**, **DESVARP** and **DESVARs** entries.

Input Data Entry: DESVARP

Description: Designates physically linked global design variable properties

Format and Example.

1	2	3	4	5	6	7	8	9	10
DESVARP	DVID	LINKID	VMIN	VMAX	VINIT	LAYERNUM	LAYRLST	LABEL	
DESVARP	1		0.01	2.0	1.0	13		NBDTOP	

Field	Contents
DVID	Design variable identification (Integer > 0)
LINKID	link identification number referring to ELIST and/or PLIST entries (Integer > 0, or blank) (Default = DVID)
VMIN	Minimum allowable value of the design variable (Real ≥ 0.0) (Default = 0.001)
VMAX	Maximum allowable value of the design variable (Real ≥ 0.0) (Default = 1000.0)
VINIT	Initial value of the design variable (Real, VMIN \leq VINIT \leq VMAX) (Default is VMIN)
LAYRNUM	Layer number if referencing a single layer of composite element(s) (Integer > 0 or blank)
LAYRLST	Set identification number of PLYLIST entries specifying a set of composite layers to be linked (Integer > 0 or blank)
LABEL	Optional user supplied label to define the design variable (Text)

Remarks:

1. The elements linked to the **DESVARP** are specified using either a **PLIST** or an **ELIST** data entry.
2. The initial element thickness or area used in the structural analysis is derived from the **VINIT** value and the property value on the associated property entry:

$$t_{init} = VINIT * property_value$$

Similarly, the minimum and maximum values are the **VMIN** and **VMAX** values of the element property are derived from:

$$t_{min} = VMIN * property_value$$

$$t_{max} = VMAX * property_value$$

3. **LAYRNUM** and **LAYRLST** are mutually exclusive.
4. Noncomposite elements may be linked to composite layers by including them in the **ELIST** and/or **PLIST** sets.

Input Data Entry: DESVARS

Description: Designates shape function linked global design variable properties.

Format and Example:

1	2	3	4	5	6	7	8	9	10
DESVARS	DVID	SHAPEID	VMIN	VMAX	VINIT	LAYERNUM	LAYRLST	LABEL	
DESVARS	1		0.01	2.0	1.0	13		INBDTOP	

Field	Contents
DVID	Design variable identification (Integer > 0)
SHAPEID	Identification number of SHAPE bulk data entries defining the shape function (Integer > 0, or blank) (Default = DVID)
VMIN	Minimum allowable value of the design variable (Real) (Default = -10^{20})
VMAX	Maximum allowable value of the design variable (Real) (Default = 10^{20})
VINIT	Initial value of the design variable (Real, $VMIN \leq VINIT \leq VMAX$) (Default is VMIN)
LAYRNUM	Layer number if referencing a single layer of composite element(s) (Integer > 0 or blank)
LAYRLST	Set identification of PLYLIST entries specifying a set of composite layers to be linked (Integer > 0 or blank)
LABEL	Optional user supplied label to define the design variable (Text)

Remarks:

1. The elements linked to the **DESVARS** are specified using either a **PLIST** or an **ELIST** data entries.
2. The initial local variables are computed from:

$$\{t_{init}\} = [P]\{VINIT\}$$

The minimum and maximum values for the local variables are taken from the **TMIN** and **TMAX** values on the property and connectivity entries, respectively.

3. **LAYRNUM** and **LAYRLST** are mutually exclusive.
4. Noncomposite elements may be linked to composite layers by including them in the referenced **SHAPE** set.

Input Data Entry: **DLAGS**

Description: This entry is used in conjunction with **RLOAD1**, **RLOAD2**, **TLOAD1** and **TLOAD2** data entries and defines time lags and phase lags as well as the set identification of the static load.

Format and Example:

1	2	3	4	5	6	7	8	9	10
DLAGS	SID	LID	TAU	PHASE	LID	TAU	PHASE		
DLAGS	5	21	0.04	20.0	10	0.0	45.0		

Field	Contents
SID	Identification number of DLAGS set (Integer > 0)
LID	Identification number of time (or frequency) independent applied load (Integer > 0)
TAU	Time delay for the designated load set (Real)
PHASE	Phase lag (in degrees) for the designated load set (Real)

Remarks:

1. One or two dynamic load sets may be defined on a single entry.
2. Refer to **RLOAD1**, **RLOAD2**, **TLOAD1** or **TLOAD2** entries for formulas which define the manner in which **TAU** and **PHASE** are used.
3. The phase parameter is used only in conjunction with **RLOAD1** and **RLOAD2** data entries.
4. The **LID** set can refer to statically applied loads as well as to additional dynamic loads input on **DLONLY** entries.
5. **TAU** and **PHASE** can be defaulted to zero, but **LID** must not be zero.

Input Data Entry: DLOAD

Description: Defines a dynamic loading condition for frequency response or transient response problems as a linear combination of load sets defined using **RLOAD1** or **RLOAD2** entries (for frequency response) or **TLOAD1** or **TLOAD2** entries (for transient response)

Format and Example:

1	2	3	4	5	6	7	8	9	10
DLOAD	SID	S	S1	L1	S2	L2	S3	L3	CONT
CONT	S4	L4		-etc-					
DLOAD	17	1.0	2.0	6	-2.0	7	2.0	8	+A
+A	-2.0	9							

Field	Contents
SID	Load set identification number (Integer > 0)
S	Scale factor (Real ≠ 0.0)
Si	Scale Factors (Real ≠ 0.0)
Li	Load set identification numbers defined via bulk data entries enumerated above (Integer > 0)

Remarks:

1. The load vector being defined by this entry is given by

$$[P] = S \sum_i S_i P_i$$

2. The **Li** must be unique.
3. **SID** must be unique from all **Li**.
4. **TLOAD1** and **TLOAD2** loads may be combined only through the use of the **DLOAD** entry.
5. **RLOAD1** and **RLOAD2** loads may be combined only through the use of the **DLOAD** entry.
6. **SID** must be unique for all **TLOAD1**, **TLOAD2**, **RLOAD1** and **RLOAD2** entries.
7. Linear load sets must be selected by a solution control command (**DLOAD = SID**).

Input Data Entry: DLONLY

Description: This entry is used in conjunction with the **RLOAD1**, **RLOAD2**, **TLOAD1** and **TLOAD2** entries and defines the point where the dynamic load is to be applied with the scale factor **A**.

Format and Example:

1	2	3	4	5	6	7	8	9	10
DLONLY	SID	P	C	A	P	C	A		
DLONLY	3	6	2	8.2	15	1	10.1		

Field	Contents
SID	Identification number of DLONLY set (Integer > 0)
P	Grid, extra point or scalar point identification number (Integer > 0)
C	Component number (1 through 6 for grid point; blank or 0 for extra points or scalar points)
A	Load factor A for the designated coordinate (Real)

Remarks:

1. One or two load factors may be defined on a single entry.
2. Refer to **RLOAD1**, **RLOAD2**, **TLOAD1** or **TLOAD2** entries for the formulas which define the load factor **A**.
3. Component numbers refer to global coordinates.
4. The **SID** field is referred to as the **DLAGS** entry.
5. The scale factor, **A**, applied to any grid/component will be the sum of all **A_i** for that degree of freedom on all **DLONLY** entries with the same **SID**.

Input Data Entry: DMI Direct Matrix Input

Description: Used to input matrix data base entities directly. Generates a real or complex matrix of the form:

$$A = \begin{bmatrix} A_{11} & A_{12} & \dots & A_{1n} \\ A_{21} & A_{22} & \dots & A_{2n} \\ \dots & \dots & \dots & \dots \\ A_{n1} & A_{n2} & \dots & A_{nn} \end{bmatrix}$$

where the elements A_{ij} may be real or complex

Format and Example:

1	2	3	4	5	6	7	8	9	10
DMI	NAME	PREC	FORM	M	N				CONT
CONT	C1	R1	A(R1, C1)	C2	R2	A(R1, C2)	A(R1+1, C2)	C3	CONT
CONT	R1	A(R1, C3)	C4	R2	A(R2, C4)				
DMI	TEST	RDP	REC	3	4				ABC
+BC	1	2	2.0	2	1	3.0	4.0	4	DEF
+EF	1	5.0	4	3	6.5				

Field	Contents
NAME	Any valid data base entity name (Text 8)
PREC	The precision of the matrix entity to be loaded. Any one of the following character strings: RSP, CSP, RDP, CDP
FORM	The form of the matrix entity to be loaded. Any one of the following REC, SYM, DIAG, IDENT, SQUARE
M	The number of rows in the matrix (Integer > 0)
N	The number of columns in the matrix (Integer > 0)
C_i	The column number of the column being loaded (Integer)
R_i	The row number of the first row in the string being loaded (Integer)
$A(R_i, C_i)$	Matrix terms (Real)

Remarks:

1. If the named entity exists, it will be flushed and reloaded. If the entity does not exist, it will be created.
2. Column and row identifiers (C_i , R_i) must always appear together although they can appear in any two contiguous fields.
3. Columns must be loaded in increasing column number order. If more than one string is to be loaded for a particular column, the C_i field must contain the same value as in the previous string. Strings must be loaded in increasing row order without overlap. Complex matrices require two real values for each matrix term. These can be split across physical entry boundaries.

Input Data Entry: DMIG**Direct Matrix Input at Grid Points**

Description: Defines structure-related direct input matrices with terms located by external grid/component values.

Format and Example:

1	2	3	4	5	6	7	8	9	10
DMIG	NAME	PREC	FORM						CONT
CONT	GCOL	CCOL	GROW	CROW	X _{ij}	Y _{ij}			CONT
CONT	GCOL	CCOL	GROW	CROW	X _{ij}	Y _{ij}			CONT

DMIG	TEST	RDP	REC						ABC
+BC	1001	4	2001	2	1.25+5				DEF
+EF	1001	4	3001	3	2.67+4	-etc-			

Field	Contents
NAME	Any valid data base entity name (Text)
PREC	The precision of the matrix entity to be loaded. Any one of the following character strings: RSP , RDP , CSP , or CDP
FORM	The form of the matrix entity to be loaded. Any one of the following: REC , SYM , DIAG , IDENT , SQUARE , TRIANG
GCOL	Grid, scalar or extra point identification for column index (Integer)
CCOL	Component number for GCOL , $0 \leq \text{CCOL} \leq 6$ if GCOL is a grid point, zero or blank for scalar or extra points. (Integer)
GROW	Grid, scalar or extra point identification for row index. (Integer)
CROW	Component number for GROW , $0 \leq \text{CROW} \leq 6$ if GROW is a grid point, zero or blank for scalar or extra points. (Integer)
X _{ij} , Y _{ij}	Matrix term. X _{ij} is real part for real or complex matrices. Y _{ij} is the imaginary part for complex matrices and is ignored for real matrices. (Real)

Remarks:

1. If the named entity exists, it will be flushed and reloaded. If the entity does not exist, it will be created.
2. The number of rows and columns will be either p-set size or g-set size depending on whether the named entity is requested by **K2PP**, **M2PP**, **B2PP** or **K2GG**, **M2GG**, etc.
3. Each non-null term in the matrix requires a continuation entry. The column index and row index values can appear any number of times on a logical entry but a fatal error will occur if the same term is entered more than once.
4. The matrix terms can be entered in any order.
5. The **TRIANG** input **FORM** implies that only the upper **or** lower triangular portion of the symmetric matrix is input. ASTROS will automatically expand the input across the diagonal.

Input Data Entry: **DYNRED** Dynamic Reduction Data

Description: Defines dynamic reduction control data.

Format and Example:

1	2	3	4	5	6	7	8	9	10
DYNRED	SID	FMAX	NVEC						
DYNRED	1	50.0							

Field	Contents
SID	Set identification number (Integer > 0)
FMAX	Highest frequency of interest (Hertz) (Real > 0 or blank)
NVEC	Number of generalized coordinates desired (Integer > 0 or blank)

Remarks:

1. Dynamic reduction data must be requested in the Solution Control packet with:
DYNRED=SID
2. The user should select either an **FMAX**, or both the **FMAX** and **NVEC** fields. **FMAX** should not be greater than necessary for the specific dynamic analysis. **NVEC**, if specified, should be significantly less than the size of the f-set to realize any computational cost savings. **NVEC** will limit dynamic reduction to using **NVEC** flexible vectors.
3. Dynamic reduction transforms the motions of the f-set to the motions of the user defined A-set plus motions of generalized coordinates created in the process. The generalized coordinates represent overall structure displacements which are approximate normal mode shapes. The generalized coordinates are identified by **SCALAR** points that are automatically generated. The **SCALAR** point identification numbers begin with 1 greater than the highest user **GRID**, **SCALAR**, or **EXTRA** point identification number.

Input Data Entry: EIGC Complex Eigenvalue Extraction Data.

Description: Specifies complex eigensolution control data.

Format and Example:

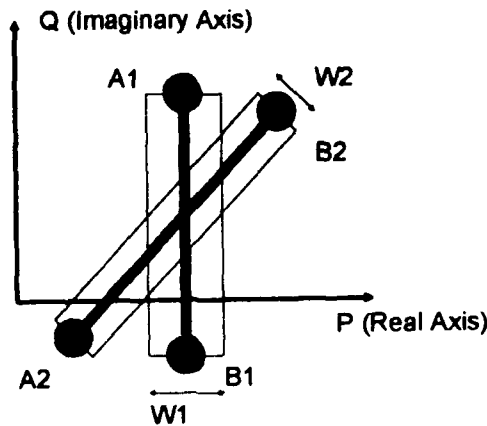
1	2	3	4	5	6	7	8	9	10
EIGC	SID	METHOD	NORM	G	C	E			abc
+bc	PA1	QA1	PB1	QB1	W1	NE1	ND1		def
+ef	PA2	QA2	PB2	QB2	W2	NE2	ND2		def

EIGC	14	INV	POINT	27		1.-8			ABC
+BC	2.0	5.6	2.0	-3.4	2.0	4	4		DEF
+EF	-5.5	-5.5	5.6	5.6	1.5	6	3		DEF

Field	Contents
SID	Set identification number (Unique integer > 0)
METHOD	Method of complex eigenvalue extraction, one of the strings INV or HESS INV - Inverse power method HESS - Upper Hessenberg method
NORM	Method for normalizing eigenvectors, one of the strings MAX or POINT MAX - Normalize to a unit value for the real part and a zero value for the imaginary part, the component having the largest magnitude. POINT - Normalize to unit value of the component G,C (defaults to MAX if point is not defined)
G	Grid or scalar point identification number (Required if and only if NORM = POINT (Integer > 0))
C	Component number (Required if and only if NORM = POINT and G is a geometric grid point) (0 < Integer ≤ 6)
E	Convergence test (Real, Default = 10 ⁻⁶)
PA _i , QA _i PB _i , QB _i	Two complex points defining a line in the complex plane (Real)
W	Width of region in complex plane (Real > 0)
NE _i	Estimated number of roots in each region (Integer > 0)
ND _i	Desired number of roots in each region (Default is 3*NE _i) (Integer > 0)

Remarks:

1. Each continuation entry defines a rectangular search region. Any number of regions may be used and they may overlap. Roots in overlapping regions will not be extracted more than once.



2. The complex eigenvalue extraction data entries must be selected in Solution Control (**CMETHOD=SID**).
3. The units of **P**, **Q**, and **W** are radians per unit time.
4. At least one continuation entry is required.
5. For the Upper Hessenberg method, **ND1** controls the number of vectors computed. Only one continuation entry is considered and the (**P**, **Q**) pairs, along with the parameters **W1** and **NE1** are ignored. All eigenvalues are computed for this method.
6. If (**P**, **Q**) pairs and parameters **W1** and **NE1** are provided, and insufficient memory exists for the Upper Hessenberg method, **ASTROS** will switch to the Inverse power method.
7. A pair (**P**, **Q**) defines a complex eigenvalue. From this pair the following may be computed:

$$f_N = \text{undamped frequency} = \frac{1}{2\pi} \sqrt{P^2 + Q^2}$$

$$\xi = \text{damping coefficient} = \frac{P}{\sqrt{P^2 + Q^2}}$$

$$f_D = \text{damped frequency} = f_N \sqrt{1 - \xi^2}$$

for lightly damped systems, **Q** is a measure of the radian frequency and **P** is a measure of the damping.

8. Parameter **W1** should be kept greater than 5 percent of the segment length **A1** to **B1** for relatively efficient processing.
9. The **SID** may be referenced directly in the **CEIG** module, or it may be entered using the **CMETHOD** option on the **BOUNDARY** command.

Input Data Entry: EIGR Real Eigenvalue Extraction Data.

Description: Defines data needed to perform real eigenvalue extraction

Format and Example:

1	2	3	4	5	6	7	8	9	10
EIGR	SID	METHOD	F1	F2	NE	ND		E	CONT
CONT	NORM	G	C						

EIGR	13	INV	1.9	15.6		12			ABC
+BC	POINT	32	4						

Field	Contents								
SID	Set identification number (Unique integer > 0)								
METHOD	Method of eigenvalue extraction, one of the character values, INV , SINV , GIV , or MGIV INV - Inverse power method SINV - Inverse power method with sturm sequence checks GIV - Given's method MGIV - Modified Given's method								
F1, F2	<table border="1"> <thead> <tr> <th>When METHOD = INV or SINV</th><th>When METHOD = GIV or MGIV</th></tr> </thead> <tbody> <tr> <td>Frequency range of interest (Real ≥ 0.0). Both must be input.</td><td>Frequency range of interest (Real > 0.0). If ND is not blank, eigenvectors are found whose natural frequencies lie in the range between F1 and F2.</td></tr> <tr> <td>Estimate of number of roots in range (Required)</td><td>Not Used</td></tr> <tr> <td>Desired number of roots. (Default is 3*NE, INV only) (Integer > 0)</td><td>Desired number of eigenvectors. (Integer > 0). If ND is blank or zero, the number of eigenvectors is determined from F1 and F2. (Default = 0)</td></tr> </tbody> </table>	When METHOD = INV or SINV	When METHOD = GIV or MGIV	Frequency range of interest (Real ≥ 0.0). Both must be input.	Frequency range of interest (Real > 0.0). If ND is not blank, eigenvectors are found whose natural frequencies lie in the range between F1 and F2 .	Estimate of number of roots in range (Required)	Not Used	Desired number of roots. (Default is 3* NE , INV only) (Integer > 0)	Desired number of eigenvectors. (Integer > 0). If ND is blank or zero, the number of eigenvectors is determined from F1 and F2 . (Default = 0)
When METHOD = INV or SINV	When METHOD = GIV or MGIV								
Frequency range of interest (Real ≥ 0.0). Both must be input.	Frequency range of interest (Real > 0.0). If ND is not blank, eigenvectors are found whose natural frequencies lie in the range between F1 and F2 .								
Estimate of number of roots in range (Required)	Not Used								
Desired number of roots. (Default is 3* NE , INV only) (Integer > 0)	Desired number of eigenvectors. (Integer > 0). If ND is blank or zero, the number of eigenvectors is determined from F1 and F2 . (Default = 0)								
NE									
ND									
E	Convergence test (Real, Default = 10 ⁻⁶)								
NORM	Method for normalizing eigenvectors, one of the character values, MASS , MAX , or POINT MASS - Normalize to unit value of the generalized mass (Default) MAX - Normalize to unit value of the largest component in the analysis set POINT - Normalize to unit value of the component defined by G,C (defaults to " MAX " if point is not defined)								
G	Grid or scalar point identification number (Required only if NORM = " POINT ") (Integer > 0)								
C	Component number (One of the integers 1-6) (Required only if NORM = " POINT " and G is a geometric grid point)								

Remarks:

1. Real eigenvalue extraction data sets must be selected in Solution Control (Method = **SID**) to be used.
2. The units of **F1** and **F2** are cycles per unit time.

3. The continuation entry is not required. **MAX** normalization is then used.
4. If **METHOD** = "GIV", all eigenvalues are found.
5. If **METHOD** = "GIV", the mass matrix for the analysis set must be positive definite. Singularities or near-singularities of the mechanism type in the mass matrix will produce poor numerical stability for the GIV method.
6. If **NORM**= **MAX**, components that are not in the analysis set may have values larger than unity.
7. If **NORM**= **POINT**, the selected component should be in the analysis set. The program uses **MAX** when it is not in the analysis set.
8. The desired number of roots (**ND**) includes all roots previously found, such as rigid body modes determined with the use of the **SUPPORT** entry.
9. The "SINV" method will actually be used for requested "INV" method.

Input Data Entry: ELEMLIST

Description: Defines a list of elements for which outputs are desired.

Format and Example:

1	2	3	4	5	6	7	8	9	10
ELEMLIST	SID	ETYPE	EID1	EID2	EID3	EID4	EID5	EID6	CONT
CONT	EID7	EID8	-etc-						CONT

ELEMLIST	100	QDMEM1	100	101	200	205			
----------	-----	--------	-----	-----	-----	-----	--	--	--

Alternate Form:

1	2	3	4	5	6	7	8	9	10
ELEMLIST	SID	ETYPE	EID1	THRU	EID2				

Field	Contents
SID	Set identification number referenced by Solution Control. (Integer > 0)
ETYPE	Character input identifying the element type. One of the following: <div><div>BAR</div><div>CONN2</div><div>ELAS</div><div>IHEX1</div><div>IHEX2</div><div>IHEX3</div><div>MASS</div><div>QDMEM1</div><div>QUAD4</div><div>ROD</div><div>SHEAR</div><div>TRIA3</div><div>TRMEM</div></div>
EID1	Element identification number (Integer > 0 or blank)

Remarks:

1. In order to be used, the **SID** must be referenced by Solution Control.
2. If the alternate form is used, **EID2** must be greater than or equal to **EID1**.
3. Nonexistent elements may be referenced and will result in no error message.
4. Any number of continuations is allowed.

Input Data Entry: **ELIST**

Description: Defines elements associated with a design variable.

Format and Example:

1	2	3	4	5	6	7	8	9	10
ELIST	LINKID	ETYPE	EID1	EID2	EID3	EID4	EID5	EID6	
CONT	EID7	EID8	EID9	-etc-					

ELIST	6	CROD	12	14	22				
-------	---	------	----	----	----	--	--	--	--

Alternate form:

1	2	3	4	5	6	7	8	9	10
ELIST	LINKID	ETYPE	EID1	THRU	EID2				

Field	Contents
LINKID	Element list identifier (Integer > 0)
ETYPE	Element type associated with this list (e.g., CROD)
EID1, EID2, EID3	Element entry identifications (Integer > 0, or blank)

Remarks:

1. Allowable **ETYPES** are: CROD, CONROD, CSHEAR, CQDMEM1, CQUAD4, CTRIA3, CTRMEM, CBAR, CELAS1, CELAS2, CONM2, CMASS1 and CMASS2.
2. If the alternate form is used, **EID2** must be greater than or equal to **EID1**.
3. All elements listed as **ELIST** entries for a particular **LINKID**, will be designed by (linked to) that design variable referencing the **ELIST LINKID**.

Input Data Entry: EPOINT Extra Point List

Description: Defines extra points of the structural model for use in dynamics problems.

Format and Example:

1	2	3	4	5	6	7	8	9	10
EPOINT	SETID	ID1	ID2	ID3	ID4	ID5	ID6	ID7	CONT
CONT	ID8	ID9	-etc-						

EPOINT	1000	3	18	1	4	16	2		
--------	------	---	----	---	---	----	---	--	--

Alternate Form:

1	2	3	4	5	6	7	8	9	10
EPOINT	SETID	ID1	THRU	ID2					

Field	Contents
-------	----------

SETID	Extra point sets identification numbers. (Integer > 0)
-------	--

IDi	Extra point identification number (Integer > 0)
-----	---

Remarks:

1. The extra point set identification is selected on the **BOUNDARY** entry. All extra points defined with this **SETID** will be used in dynamic analyses in the boundary condition.
2. All extra point identification numbers must be unique with respect to all other structural and scalar points.
3. This entry is used to define coordinates used in transfer function definitions (see **TF** entry) and Direct Matrix input.
4. If the alternate form is used, **ID2** must be greater than or equal to **ID1**.

Input Data Entry: FFT

Description: Defines parameters for controlling the Fast Fourier Transformation (**FFT**) during time domain response analysis.

Format and Example:

1	2	3	4	5	6	7	8	9	10
FFT	SID	TIME	NT	RDELTF	RF	FRIM	OTYPE	FLIM	
FFT	3	20.	1024	1.0					

Field	Contents
SID	FFT set identification number (Integer > 0)
TIME	Length of time period to be analyzed (Real > 0.0)
NT	Number of time points to be used for the FFT (Integer ≥ 2)
RDELTF	Ratio of incremental frequency (del F) to $1 / T$. See remarks 4 and 6. (Default = 1.0, Real > 0.0)
RF	Ratio of total frequency duration (F) to $NT / 2 * T$. See remarks 5 and 6. (Default = 1.0, Real > 0.0)
FRIM	Frequency response interpolation method. Character string LINEAR or CUBIC . Default is LINEAR .
OTYPE	Type of response to be output. Character string TIME , FREQ or BOTH . Default is TIME .
FLIM	Frequency load interpolation method. Character string LINEAR or CUBIC . Default is LINEAR .

Remarks:

1. **SID** must be selected by a **FFT** option on a **TRANSIENT** command in solution control.
2. **TIME** is the period for periodic dynamic loads defined in the time domain. For non-periodic loads, T is the total time duration of the excitation plus any quiet portion desired for response decay. T may be larger than the time duration defined by **TLOAD1** or **TLOAD2** data, in which case the forcing function will be automatically set to zero for the additional time.
3. **NT** should be a power of 2; i.e., $NT = 2^m$, $m = 1, 2, \dots$; or $NT = 2, 4, 8, \dots$. If **NT** is not a power of 2, it will be automatically set to the next highest power of 2 value.
4. The incremental frequency, ΔF , required by the **FFT** algorithm, is $1 / T$. The value of ΔF may be adjusted by the user with the **RDELTF** factor. However, the most accurate results are normally obtained with the default case of **RDELTF** = 1.0.
5. The frequency duration required by the **FFT** algorithm is $F = NT / 2 * T$. This is the frequency duration used when the default value of **RF** = 1.0 is used. If **RF** < 1.0, the response between **RF** and 1.0 is set to zero when using the inverse Fourier transform to compute time domain responses.
6. The frequency list used in the frequency response calculations is generated using a constant incremental frequency of $\text{del } F = \text{RDELTF} * \Delta F$, and the total frequency duration is $F = \text{RF} * F$.

Input Data Entry: **FLFACT** **Aerodynamic Physical Data**

Description: Used to specify density ratios, velocity lists, Mach numbers, and reduced frequencies for **FLUTTER** analysis.

Format and Example:

1	2	3	4	5	6	7	8	9	10
FLFACT	SID	F1	F2	F3	F4	F5	F6	F7	CONT
CONT	F8	F9	-etc-						
FLFACT	97	.3	.7	3.5					

Field	Contents
SID	Set identification number (Integer > 0).
F1	Aerodynamic factor (Real).

Remarks:

1. Only the factors selected by a **FLUTTER** data entry will be used.
2. Embedded blank fields are forbidden.
3. Parameters must be listed in the order in which they are to be used within the looping of **FLUTTER** analysis.
4. All **FLFACT** entries having the same **SETID** will be treated as a single set.

Input Data Entry: **FLUTTER** Aerodynamic **FLUTTER** Data

Description: Defines data needed to perform **FLUTTER** analysis.

Format and Example:

1	2	3	4	5	6	7	8	9	10
FLUTTER	SID	METHOD	DENS	MACH	VEL	MLIST	KLIST	EFFID	CONT
CONT	SYMxz	SYMxy	EPS	CURFIT					
FLUTTER	19	PK	119	0.85	319	10			

Field	Contents						
SID	Set identification number (Integer > 0)						
METHOD	FLUTTER analysis method, " PK " for PK -method (Text)						
DENS	Identification number of an FLFACT set specifying density ratios to be used in FLUTTER analysis (Integer > 0).						
MACH	Mach number to be used in the FLUTTER analysis (Real ≥ 0.0)						
VEL	Identification number of an FLFACT set specifying velocities to be used in the FLUTTER analysis. (Integer > 0).						
MLIST	Identification number of a SET1 set specifying a list of normal modes to be <i>omitted</i> from the FLUTTER analysis (Integer > 0, or blank).						
KLIST	Identification number of an FLFACT set specifying a list of <i>hard point</i> reduced frequencies for the given Mach number for use in the FLUTTER analysis (See Remark 4) (Integer ≥ 0 , or blank)						
EFFID	Identification number of a CONEFF set specifying control surface effectiveness values (See Remark 5) (Integer ≥ 0 , or blank)						
SYMxz, SYMxy	Symmetry flags associated with the aerodynamics (See Remark 6) (Integer) <table> <tr> <td>+1</td><td>Symmetric</td></tr> <tr> <td>0 or blank</td><td>Asymmetric</td></tr> <tr> <td>-1</td><td>Antisymmetric</td></tr> </table>	+1	Symmetric	0 or blank	Asymmetric	-1	Antisymmetric
+1	Symmetric						
0 or blank	Asymmetric						
-1	Antisymmetric						
EPS	Convergence parameter for FLUTTER eigenvalue (Real, Default = 10^{-5})						
CURFIT	Type of curve fit to be used in the PK FLUTTER analysis. One of LINEAR , QUAD , CUBIC , or ORIG (See Remarks 7, 8, and 9) (Text, Default = LINEAR)						

Remarks:

1. The **FLUTTER** data entry must be selected in the Solution Control packet. Only those Mach numbers and symmetries selected in Solution will be processed in the **UNSTEADY** aerodynamic preface.
2. The density is given by $\rho \times \rho_{ref}$, where ρ_{ref} is the reference value given on the **AERO** Bulk Data entry, and ρ is the density ratio from the **FLFACT** entry.
3. If the **MLIST** is blank or zero, all computed normal modes will be retained in the **FLUTTER** analysis.

4. If the **KLIST** is blank or zero, all "hard point" k values (those on the **MKAEROi** entries) associated with the Mach number/symmetries on the **FLUTTER** entry will be used in the interpolation of the aerodynamics. Specifying a subset may be used to improve the **ORIG** interpolation. Those hard point k values nearest in value to those listed on the **FLFACT** will be used. No duplicate hard point k's will be used and no errors will be printed.
5. If the **EFFID** is blank or zero, no effectiveness corrections will be made.
6. The symmetry flags are used to select the appropriate unsteady aerodynamic matrices generated from the list on the **MKAEROi** entries.
7. The **LINEAR**, **QUAD**, and **CUBIC** fits are separate first, second and third order, respectively, fits of the real and complex terms of the generalized aerodynamic matrix at each hard point k. Only the closest 2, 3 or 4, respectively, k's are utilized for each fit and **LINEAR** fitting is used off the ends of the hard point **KLIST**. The program automatically reduces the order of the fit if too few points are available for the higher order fit (e.g., **CUBIC** becomes **QUAD** if only 3 k's are used in the **KLIST**) (Refer to the Version 9.0 Release Notes for more information).
8. The **ORIGINAL** fit (documented in the Theoretical Manual) is a cubic fit over all the hard point k's. Its use is not recommended since it tends to experience numerical problems for any but small k ranges and small numbers of k's.
9. For all fitting options, the generalized aerodynamic matrices are normalized by the hard point k value before fitting, as documented on the Theoretical Manual.

Input Data Entry: **FORCE** Static Load

Description: Defines a static load at a grid point by specifying a vector.

Format and Example:

1	2	3	4	5	6	7	8	9	10
FORCE	SID	G	CID	F	N1	N2	N3		
FORCE	2	5	6	2.9	0.0	1.0	0.0		

Field	Contents
SID	Load set identification number (Integer > 0)
G	Grid point identification number (Integer > 0)
CID	Coordinate system identification number (Integer ≥ 0, or blank) (Default = 0)
F	Scale factor (Real)
Ni	Components of a vector measured in the coordinate system defined by CID (Real; must have at least one nonzero component)

Remarks:

1. The static load applied to grid point **G** is given by

$$\{F\} = F \{N\}$$

where **{N}** is the vector defined in Fields 6, 7 and 8.

2. A **CID** of zero references the basic coordinate system.

Input Data Entry: **FORCE1** Static Load, Alternate Form 1

Description: Used to define a static load by specification of a value and two grid points which determine the direction.

Format and Example:

1	2	3	4	5	6	7	8	9	10
FORCE1	SID	G	F	G1	G2				
FORCE1	6	13	-2.93	16	13				

Field	Contents
SID	Load set identification number (Integer > 0)
G	Grid point identification number (Integer > 0)
F	Value of load (Real)
G1	Grid point identification numbers (Integer > 0; G1 ≠ G2)

Remarks:

1. The direction of the force is determined by the vector from **G1** to **G2**.

Input Data Entry: FREQ

Description: Defines a set of frequencies to be used in the solution of frequency response problems.

Format and Example:

1	2	3	4	5	6	7	8	9	10
FREQ	SID	F1	F2	F3	F4	F5	F6	F7	CONT
CONT	F8	F9	-etc-						CONT
FREQ	3	2.98	3.05	17.9	21.3	25.6	28.8	31.2	ABC
+BC	29.2	22.4	19.3						

Field	Contents
SID	Frequency set identification number (Integer > 0)
F1	Frequency value (Real ≥ 0.0)

Remarks:

1. The units for the frequencies are cycles per unit time.
2. Frequency sets must be selected by the Solution Control (**FSTEP=SID**) to be used.
3. All **FREQ**, **FREQ1** and **FREQ2** entries with the same frequency set identification numbers will be used. Duplicate frequencies will be ignored. **f_N** and **f_{N-1}** are considered duplicated if

$$|f_N - f_{N-1}| < 10^{-5} * (f_{MAX} - f_{MIN})$$

Input Data Entry: FREQ1

Description: Defines a set of frequencies to be used in the solution of frequency response problems by specification of a starting frequency, frequency increment, and number of increments desired.

Format and Example:

1	2	3	4	5	6	7	8	9	10
FREQ1	SID	F1	DF	NDF					
FREQ1	6	2.9	0.5	13					

Field	Contents
SID	Frequency set identification number (Integer > 0)
F1	First frequency in set (Real ≥ 0.0)
DF	Frequency increment (Real > 0.0)
NDF	Number of frequency increments (Integer > 0)

Remarks:

1. The units for the frequency **F1** and the frequency increment **DF** are cycles per unit time.
2. The frequencies defined by this entry are given by:
$$f_i = F1 + (i-1) DF, \quad i = 1, NDF + 1$$
3. Frequency sets must be selected by the Solution Control (**FSTEP**=**SID**) to be used.
4. All **FREQ**, **FREQ1** and **FREQ2** entries with the same frequency set identification numbers will be used. Duplicate frequencies will be ignored. **f_N** and **f_{N-1}** are considered duplicated if

$$|f_N - f_{N-1}| < 10^{-6} * (f_{MAX} - f_{MIN})$$

Input Data Entry: FREQ2

Description: Defines a set of frequencies to be used in the solution of frequency response problems by specification of a starting frequency, final frequency, and number of logarithmic increments desired.

Format and Example:

1	2	3	4	5	6	7	8	9	10
FREQ2	SID	F1	F2	NF					
FREQ2	6	1.0	8.0	6					

Field	Contents
SID	Frequency set identification number (Integer > 0)
F1	First frequency (Real > 0.0)
F2	Last frequency (Real > 0.0; F2 > F1)
NF	Number of logarithmic intervals (Integer > 0)

Remarks:

1. The units for the frequencies **F1** and **F2** are cycles per unit time.
2. The frequencies defined by this entry are given by:

$$f_i = F_1 e^{i-1} d$$

where,

$$d = \left(\frac{1}{NF} \right) \ln \left(\frac{f_2}{f_1} \right)$$

For the example shown, the list of frequencies will be 1.0, 1.4142, 2.0, 2.8284, 4.0, 5.6569 and 8.0 cycles per unit time.

3. Frequency sets must be selected by the Solution Control (**FSTEP=SID**) to be used.
4. All **FREQ**, **FREQ1** and **FREQ2** entries with the same frequency set identification numbers will be used. Duplicate frequencies will be ignored. **fN** and **fN-1** are considered duplicated if:

$$| f_N - f_{N-1} | < 10^{-5} * (f_{MAX} - f_{MIN})$$

Input Data Entry: **FREQLIST**

Description: Defines a list of frequencies for which outputs are defined.

Format and Example:

1	2	3	4	5	6	7	8	9	10
FREQLIST	SID	FREQ1	FREQ2	FREQ3	FREQ4	FREQ5	FREQ6	FREQ7	CONT
CONT	FREQ8	FREQ9	-etc-						
FREQLIST	100	10.0	20.0	50.0	100.0				

Field	Contents
SID	Set identification number referenced by Solution Control (Integer > 0)
FREQi	Frequency (in Hertz) at which outputs are desired (Real)

Remarks:

1. In order to be used, the **SID** must be referenced by Solution Control.
2. The nearest frequency to **FREQ**, either above or below, which was used in the Frequency Response analysis will be used to satisfy the output requests.
3. Any number of continuations is allowed.

Input Data Entry **GDVLIST** Global Design Variable List

Description: Defines a list of global design variables for which outputs are desired.

Format and Example:

1	2	3	4	5	6	7	8	9	10
GDVLIST	SID	GDVID1	GDVID23	GDVID3	GDVID4	GDVID5	GDVID6	GDVID7	CONT
CONT	GDVID8	GDVID9	-etc-						

GDVLIST	100	1	2	3	5	7	9		
---------	-----	---	---	---	---	---	---	--	--

Alternate Form

1	2	3	4	5	6	7	8	9	10
GDVLIST	SID	GDVID1	THRU	GDVID2					

Field	Contents								
SID	Set identification number referenced by Solution Control (Integer > 0)								
GDVID	Global design variable identification number (Integer > 0 or blank)								

Remarks:

1. In order to be used, the **SID** must be referenced by Solution Control.
2. If the alternate form is used, **GDVID2** must be greater than or equal to **GDVID1**.
3. Nonexistent global design variables may be referenced and will result in no error message.
4. Any number of continuations is allowed, except when using the alternate form, which allows no continuations.

Input Data Entry GPWG Weight Generator Data

Description: Contains definition of the location about which to perform grid point weight generation

Format and Example:

1	2	3	4	5	6	7	8	9	10
GPWG	GID/XO	YO	ZO						
GPWG	10								

Field	Contents
GID	Grid point identification of the GPWG reference point (Integer)
XO	X component of basic coordinates of the reference point (Real)
YO	Y component of basic coordinates of the reference point (Real)
ZO	Z component of basic coordinates of the reference point (Real)

Remarks:

1. Either a grid point identification number or the basic x, y, z components of the reference point may be given.
2. If no **GPWG** data entry exists, the grid point weight generation will be computed about the origin of the basic coordinate system.
3. If more than one **GPWG** entry exists, the first one appearing in the sorted bulk data echo will be used.

Input Data Entry: **GRAV** Gravity Vector

Description: Used to define gravity vectors for use in determining gravity loading for the structural model.

Format and Example:

1	2	3	4	5	6	7	8	9	10
GRAV	SID	CID	G	N1	N2	N3			
GRAV	1	3	32.2	0.0	0.0	-1.0			

Field	Contents
SID	Set identification number (Integer > 0)
CID	Coordinate system identification number (Integer ≥ 0)
G	Gravity vector scale factor (Real ≠ 0.0)
Ni	Gravity vector components (Real; at least one nonzero component)

Remarks:

1. The gravity vector is defined by $\{g\} = G\{N_i\}$. The direction of $\{g\}$ is the direction of free fall.
2. A CID of zero references the basic coordinate system.
3. Gravity loads may be combined with "simple loads" (e.g., **FORCE**, **MOMENT**) by specification on a **LOAD** entry or by **GRAV = SID**. Gravity loads with the same **SID** as simple load entries will not be used unless referenced by one of these methods.
4. Load sets must be selected in Solution Control to be used.
5. The units of **G** should be length/sec² in consistent length units.

Input Data Entry: **GRDSET** Grid Point Default

Description: Defines default options for Fields 3, 7, and 8 of all GRID entries.

Format and Example:

1	2	3	4	5	6	7	8	9	10
GRDSET		CP				CD	PS		
GRDSET		16				32	3456		

Field	Contents
CP	Identification number of coordinate system in which the location of the grid point is defined (Integer ≥ 0)
CD	Identification number of coordinate system in which the displacements are measured at grid point (Integer ≥ 0)
PS	Permanent single-point constraints associated with grid point (any of the digits 1 through 6 with no embedded blanks) (Integer ≥ 0)

Remarks:

1. The contents of Fields 3, 7, or 8 of this entry are assumed for the corresponding fields of any GRID entry whose Field 3, 7, and 8 are blank. If any of these fields on the GRID entry are blank, the default option defined by this entry occurs for that field. If no permanent single-point constraints are desired or one of the coordinate systems is basic, the default may be overridden on the GRID entry by making one of the Fields 3, 7, or 8 zero (rather than blank). Only one **GRDSET** entry may appear in the user's Bulk Data packet.
2. The primary purpose of this entry is to minimize the burden of preparing data for problems with a large amount of repetition (e.g., two-dimensional pinned-joint problems).
3. At least one of the entries **CP**, **CD**, or **PS** must be nonzero.

Input Data Entry: **GRID** Grid Point

Description: Defines the location of a geometric grid point of the structural model, the directions of its displacement, and its permanent single-point constraints.

Format and Example:

1	2	3	4	5	6	7	8	9	10
GRID	ID	CP	X1	X2	X3	CD	PS		
GRID	2	3	1.0	2.0	3.0		315		

Field	Contents
ID	Grid point identification number (Integer > 0)
CP	Identification number of coordinate system in which the location of the grid point is defined (Integer > 0 or blank)
X1	Location of the grid point in coordinate system CP (Real)
CD	Identification number of coordinate system in which displacements, degrees of freedom, constraints, and solution vectors are defined at the grid point (Integer > 0 or blank)
PS	Permanent single-point constraints associated with grid point (any of the digits 1-6 with no embedded blanks) (Integer > 0 or blank)

Remarks:

1. All grid point identification numbers must be unique with respect to all other structural and scalar points.
2. The meaning of **x1**, **x2**, and **x3** depend on the type of coordinate system, **CP**, as follows:

TYPE	X1	X2	X3
Rectangular	X	Y	Z
Cylindrical	R	θ (deg)	Z
Spherical	R	θ (deg)	ϕ (deg)

Also see **CORDi j** entry descriptions.

3. The collection of all **CD** coordinate systems defined on all **GRID** entries is called the Global Coordinate System. All degrees-of-freedom, constraints, and solution vectors are expressed in the Global Coordinate System.

Input Data Entry: GRIDLIST

Description: Defines a list of points at which outputs are desired.

Format and Example:

1	2	3	4	5	6	7	8	9	10
GRIDLIST	SID	GID1	GID2	GID3	GID4	GID5	GID6	GID7	CONT
CONT	GID8	GID9	-etc-						

GRIDLIST	100	1001	1010	1020					
----------	-----	------	------	------	--	--	--	--	--

Alternate Form:

1	2	3	4	5	6	7	8	9	10
GRIDLIST	SID	GID1	THRU	GID2					

Field	Contents
SID	Set identification number referenced by Solution Control (Integer > 0)
GID _i	Grid, scalar or extra point id at which outputs are desired (Integer > 0)

Remarks:

1. In order to be used, the **SID** must be referenced by Solution Control.
2. If the alternate form is used, **GID2** must be greater than or equal to **GID1**.
3. Nonexistent points may be referenced and will result in no error message.
4. Any number of continuations is allowed.

Input Data Entry: GUST**Aerodynamic Gust Load Description****Description:** Defines a stationary vertical gust for use in aeroelastic analysis.**Format and Example:**

1	2	3	4	5	6	7	8	9	10
GUST	SID	GLOAD	WG	XO	V	QDP	MACH		CONT
CONT	SYMXZ	SYMXY							

GUST	133	61	1.0	0.	1.+4	13.5	0.9		ABC
+BC	1	0							

Field	Contents
SID	Gust set identification number (Integer > 0)
GLOAD	The SID of a TLOAD or RLOAD data entry which defines the time or frequency dependence (Integer > 0)
WG	Scale factor (gust velocity/forward velocity) for gust velocity (Real ≠ 0.)
XO	Location of reference plane in aerodynamic coordinates (Real).
V	Velocity of vehicle (Real > 0.0)
QDP	Dynamic pressure (Real > 0.0)
MACH	Mach number (Real ≥ 0.0)
SYMXZ, SYMXY	Symmetry flags (Integer)

Remarks:

1. The **GUST** entry is selected as a discipline option for **FREQUENCY** or **TRANSIENT** in Solution Control.
2. The gust angle is in the +z direction of the aerodynamic coordinate system. The value is,

$$WG = T \left[t - \frac{x - x_o}{v} \right]$$

where T is the tabular function.

3. The symmetry flags will be used to select the appropriate unsteady aerodynamic matrices from the list of m-k pairs for each symmetry option given on the **MKAEROi** entries.

Input Data Entry: IC Transient Initial Condition

Description: Defines values for the initial conditions of coordinates used in transient analysis. Both displacement and velocity values may be specified at independent coordinates of the structural model.

Format and Example:

1	2	3	4	5	6	7	8	9	10
IC	SID	G	C	UO	VO				
IC	1	3	2	5.0	-6.0				

Field	Contents
SID	Set identification number (Integer > 0)
G	Grid or scalar or extra point identification number (Integer > 0)
C	Component number (blank or zero for scalar or extra points, any one of the digits 1 through 6 for a grid point)
UO	Initial displacement value (Real)
VO	Initial velocity value (Real)

Remarks:

1. Transient initial condition sets must be selected in the Solution Control (**IC=SID**) to be used.
2. If no **IC** set is selected, all initial conditions are assumed zero.
3. Initial conditions for coordinates not specified on **IC** entries will be assumed zero.
4. Initial conditions may be used only in direct formulation. In a modal formulation the initial conditions are all zero.

Input Data Entry ITERLIST Iteration List

Description: Defines a list of iteration steps for which outputs are desired.

Format and Example:

1	2	3	4	5	6	7	8	9	10
ITERLIST	SID	ITER	ITER	ITER	ITER	ITER	ITER	ITER	CONT
CONT	ITER	ITER	-etc-						

ITERLIST	100	1	2	3	5	7	9		
----------	-----	---	---	---	---	---	---	--	--

Alternate Form:

1	2	3	4	5	6	7	8	9	10
ITERLIST	SID	ITER	THRU	ITER					

Field

Contents

SID Set identification number referenced by Solution Control.
 (Integer > 0)

ITER Iteration step number. (Integer > 0 or blank)

Remarks:

1. In order to be used, the **SID** must be referenced by Solution Control.
2. Nonexistent iteration steps may be referenced and will result in no error message.
3. Any number of continuations is allowed, except when using the alternate form, which allows no continuations.

Input Data Entry: **JSET** Select Coordinates for the j-set

Description: Defines coordinates (degrees of freedom) that the user desires to use in the computation of inertia relief mode shape in Dynamic Reduction.

Format and Example:

1	2	3	4	5	6	7	8	9	10
JSET	SETID	ID	C	ID	C	ID	C		CONT
CONT	ID	C	ID	C	-etc-				
JSET	16	2	23	3516					

Field	Contents
SETID	The set identification number of the INERTIA set. (Integer > 0)
ID	Grid or scalar point identification number (Integer > 0).
C	Component number, zero or blank for scalar points, any unique combination of the digits 1 through 6 for grid points. (Integer)

Remarks:

- Coordinates specified on this entry form members of a mutually exclusive set. They may not be specified on other entries that define mutually exclusive sets.
- When **JSET** and/or **JSET1** entries are present, all degrees of freedom not otherwise constrained will be placed on the o-set.
- Use of **JSET** in dynamic reduction:
 - JSET** defines the structural/nonstructural j-set degrees of freedom (inertia relief shapes). An alternate input format is provided by the **JSET1** entry.
 - The **SID** is selected by the Solution Control Command **BOUNDARY INERTIA = n**.
 - Use "0" as the grid point identification number to select the origin of the basic coordinate system as the j-set degrees of freedom.
- Any number of continuations are allowed.

Input Data Entry: **JSET1** Select Coordinates for the j-set, Alternate Form

Description: Defines coordinates (degrees of freedom) that the user desires to use in the computation of inertia relief mode shape(s) in Dynamic Reduction.

Format and Example:

1	2	3	4	5	6	7	8	9	10
JSET1	SETID	C	GID1	GID2	GID3	GID4	GID5	GID6	CONT
CONT	GID7	GID8	-etc-						

JSET1	345	2	1	3	10	9	6		ABC
+bc	7	8							

Alternate Form:

1	2	3	4	5	6	7	8	9	10
JSET1	SETID	C	GID1	THRU	GID2				

Field	Contents
SETID	The INERTIA set identification number
C	Component number (any unique combination of the digits 1 through 6 (with no embedded blanks) when point identification numbers are grid points; must be blank or zero if point identification numbers are scalar points.
GID _i	Grid or scalar point identification numbers (Integer > 0).

Remarks:

- Coordinates specified on this entry form members of a set that is exclusive from other sets defined by bulk data entries.
- When **JSET** and/or **JSET1** entries are present, all degrees of freedom not otherwise constrained will be placed in the o-set.
- If the alternate form is used, all points in the sequence **ID1** through **ID2** are required to exist and **ID2** must be greater than or equal to **ID1**.
- Use of **JSET1** in dynamic reduction:
 - JSET1** defines the structural and nonstructural j-set degrees of freedom (inertia relief shapes). An alternate input format is provided by the **JSET** entry.
 - The **SID** is selected by Solution Control Command **BOUNDARY INERTIA = n**.
 - Use "0" as the grid point identification number to select the origin of the basic coordinate system as the j-set degrees freedom.

Input Data Entry LDVLIST Local Design Variable List

Description: Defines a list of local design variables for which outputs are desired.

Format and Example:

1	2	3	4	5	6	7	8	9	10
LDVLIST	SID	ETYPE	LAYER	EID1	EID2	EID3	EID4	EID5	CONT
CONT	EID6	EID7	-etc-						
LDVLIST	100	QUAD4	2	100	100	200	300	700	

Alternate Form

1	2	3	4	5	6	7	8	9	10
LDVLIST	SID	ETYPE	LAYER	EID1	THRU	EID2			

Field	Contents												
SID	Set identification number referenced by Solution Control. (Integer > 0)												
ETYPE	Character input identifying the element type. One of the following: <table><tr><td>BAR</td><td>MASS</td><td>ROD</td><td>TRMEM</td></tr><tr><td>CONM2</td><td>QDMEM1</td><td>SHEAR</td><td></td></tr><tr><td>ELAS</td><td>QUAD4</td><td>TRIA3</td><td></td></tr></table>	BAR	MASS	ROD	TRMEM	CONM2	QDMEM1	SHEAR		ELAS	QUAD4	TRIA3	
BAR	MASS	ROD	TRMEM										
CONM2	QDMEM1	SHEAR											
ELAS	QUAD4	TRIA3											
LAYER	Layer number if element is composite laminate. (Integer > 0 or blank)												
EID _i	Element identification number. (Integer > 0 or blank)												

Remarks:

1. In order to be used, the **SID** must be referenced by Solution Control.
2. If the alternate form is used **EID2** must be greater than or equal to **EID1**.
3. Nonexistent elements may be referenced and will result in no error message.
4. If a layer number is omitted for a composite laminate element then all layers in that element will be selected.
5. Any number of continuations is allowed.

Input Data Entry: **LOAD** Static Load Combination (Superposition)

Description: Defines a static load as a linear combination of load sets defined using **FORCE**, **MOMENT**, **FORCE1**, **MOMENT1**, **PLOAD**, and **GRAV** entries.

Format and Example:

1	2	3	4	5	6	7	8	9	10
LOAD	SID	S	S1	L1	S2	L2	S3	L3	CONT
CONT	S4	L4							

LOAD	101	-0.5	1.0	3	6.2	4	4.5	10	ABC
+BC	2.3	115							

Field	Contents
SID	Load set identification number (Integer >0)
S	Scale factor (Real ≠ 0.0)
Si	Scale factors (Real ≠ 0.0)
Li	Load set identification numbers defined via data entry types enumerated above (Integer > 0)

Remarks:

1. The load vector defined is given by

$$P = S \sum S_i P_{Li}$$
2. The **Li** must be unique. The remainder of the physical entry containing the last entry must be blank.
3. Load sets must be selected in the Solution Control if they are to be applied to the structural model.
4. A **LOAD** entry may not reference a set identification number defined by another **LOAD** entry.

Input Data Entry: MAT1**Material Property Definition, Form 1**

Description: Defines the material properties for linear, temperature-independent, isotropic materials

Format and Example:

1	2	3	4	5	6	7	8	9	10
MAT1	MID	E	G	NU	RHO	A	TREF	GE	CONT
CONT	ST	SC	SS	MCSID					
MAT1	17	3.+7		0.33	4.28	6.5-6	5.37-6	0.23	ABC
+B	20.+4	15.+4	12.+4						

Field	Contents
MID	Material identification number (Integer >0)
E	Young's modulus (Real > 0.0, or blank)
G	Shear modulus (Real or blank)
NU	Poisson's ratio ($-1.0 < \text{Real} \leq 0.5$ or blank)
RHO	Mass density (Real ≥ 0.0)
A	Thermal expansion coefficient (Real)
TREF	Thermal expansion reference temperature (Real)
GE	Structural element damping coefficient (Real)
ST, SC, SS	Stress limits for tension, compression, and shear (Real). (Used to compute margins of safety in certain elements).
MCSID	Material Coordinate System identification number (Integer > 0 or blank).

Remarks:

1. The material identification number must be unique for all **MAT1**, **MAT2**, **MAT8**, and **MAT9** bulk data entries.
2. The mass density, **RHO**, will be used to automatically compute mass for all structural elements.
3. Weight density may be used in Field 6 if the value $1/g$ is entered on the **CONVERT** entry where g is the acceleration of gravity.
4. Either **E** or **G** must be specified (i.e., nonblank).
5. If any one of **E**, **G**, or **NU** is blank, it will be computed to satisfy the identity $E = 2 * (1+NU) * G$; otherwise, values supplied by the user will be used.
6. If **E** and **NU** or **G** and **NU** are both blank, they will both be given the values 0.0.
7. Implausible data on one or more **MAT1** entries will result in a warning message. Implausible data is defined as any of $E < 0.0$ or $G < 0.0$ or $NU > 0.5$ or $NU < 0.0$ or $|1 - E/2(1+NU)G| > 0.01$ except for cases covered by Remark 6.
8. It is strongly recommended that only two of the three values **E**, **G**, and **NU** be input. The three values may be input independently on the **MAT2** entry.

Input Data Entry: MAT2**Material Property Definition, Form 2**

Description: Defines the material properties for linear, temperature-independent, anisotropic materials for two-dimensional elements.

Format and Example:

1	2	3	4	5	6	7	8	9	10
MAT2	MID	G11	G12	G13	G22	G23	G33	RHO	CONT
CONT	A1	A2	A12	TO	GE	ST	SC	SS	CONT
CONT	MCSID								

MAT2	13	6.2+3			6.2+3		5.1+3	0.056	ABC
+BC	6.5-6	6.5-6		-500.0	0.002	20.+5			

Field	Contents
MID	Material identification number (Integer > 0)
G _{ij}	The material property matrix (Real)
RHO	Mass density (Real ≥ 0.0)
A _i	Thermal expansion coefficient vector (Real)
TO	Thermal expansion reference temperature (Real)
GE	Structural element damping coefficient (Real)
ST, SC, SS	Stress limits for tension, compression, and shear (Real). (Used to compute margins of safety in certain elements).
MCSID	Material Coordinate System identification number (Integer > 0 or blank).

Remarks:

1. The material identification numbers must be unique for all **MAT1**, **MAT2**, **MAT8**, and **MAT9** bulk data entries.
2. The mass density, **RHO**, will be used to automatically compute mass for all structural elements.
3. The convention for the G_{ij} in Fields 3 through 8 are represented by the matrix relationship

$$\begin{Bmatrix} \sigma_1 \\ \sigma_2 \\ \tau_{12} \end{Bmatrix} = \begin{bmatrix} G_{11} & G_{12} & G_{13} \\ G_{12} & G_{22} & G_{23} \\ G_{13} & G_{23} & G_{33} \end{bmatrix} \begin{Bmatrix} \epsilon_1 \\ \epsilon_2 \\ \gamma_{12} \end{Bmatrix} - (T - T_0) \begin{Bmatrix} A_1 \\ A_2 \\ A_{12} \end{Bmatrix}$$

4. 2x2 matrices (for example, transverse shear) use elements **G11**, **G12**, and **G22**. For this case, **G33** must be blank.
5. If the **MAT2** entry is referenced by the **PCOMP** entry, the transverse shear flexibility for the referenced laminae is zero.
6. Unlike the **MAT1** entry, data from the **MAT2** entry are used directly, without adjustment of equivalent **E**, **G**, or **NU** values.

Input Data Entry: MAT8**Material Property Definition, Form 8****Description:** Defines the material property for an orthotropic material.**Format and Example:**

1	2	3	4	5	6	7	8	9	10
MAT8	MID	E1	E2	NU12	G12	G1, Z	G2, Z	RHO	CONT
CONT	A1	A2	TREF	Xt	Xc	Yt	Yc	S	CONT
CONT	GE	F12							

MAT8	171	30.+6	1.+6	0.3	2.+6	3.+6	1.5+6	0.056	+ABC
+BC	28.-6	1.5-6	155.0	1.+4	1.5+4	2.+2	8.+2	1.+3	+DEF
+EF	1.-4								

Field	Contents
MID	Material identification number (Integer > 0)
E1	Modulus of elasticity in longitudinal direction (also defined as fiber direction or 1-direction) (Real ≠ 0.0)
E2	Modulus of elasticity in lateral direction (also defined as matrix direction or 2-direction) (Real ≠ 0.0)
NU12	Poisson's ratio $\left[\frac{(\epsilon_2)}{(\epsilon_1)} \text{ for uniaxial loading in 1-direction} \right]$. Note that $NU21 = \frac{(\epsilon_1)}{(\epsilon_2)}$ for uniaxial loading in 2-direction is related to NU12, E1, E2 by the relation $NU12 \cdot E2 = NU21 \cdot E1$. (Real)
G12	In-plane shear modulus (Real > 0.0)
G1,Z	Transverse shear modulus for shear in 1-Z plane (Real > 0.0 or blank) (default implies infinity)
G2,Z	Transverse shear modulus for shear in 2-Z plane (Real > 0.0 or blank) (default implies infinity)
RHO	Mass density (Real ≥ 0.0)
A1	Thermal expansion coefficient in the 1-direction (Real)
A2	Thermal expansion coefficient in the 2-direction (Real)
TREF	Thermal expansion reference temperature (Real)
Xt, Xc	Allowable stresses in tension and compression, respectively, in the longitudinal direction. Required if failure index is desired. (Real ≥ 0.0) (Default value for Xc is Xt)
Yt, Yc	Allowable stresses in tension and compression, respectively, in the transverse direction. Required if failure index is desired. (Real ≥ 0.0) (Default value for Yc is Yt)
S	Allowable stress for in-plane shear (Real ≥ 0.0)
GE	Structural damping coefficient (Real)

F12

Interaction term in the tensor polynomial theory of Tsai-Wu (Real). Required if failure index or stress constraint by Tsai-Wu theory is desired and if value of F12 is different from 0.0.

Remarks:

1. If G_1, Z and G_2, Z values specified as zero, or are not supplied, transverse shear flexibility calculations will not be performed.
2. An approximate value for G_1, Z and G_2, Z is the in-plane shear modulus G_{12} . If test data are not available to accurately determine G_1, Z and G_2, Z for the material and transverse shear calculations are deemed essential, the value of G_{12} may be supplied for G_1, Z and G_2, Z .
3. x_t, x_c, y_t, y_c and ss are used for composite element failure calculations when requested in the **PT** field of the **PCOMP1** entry.

Input Data Entry: MAT9 Material Property Definition, Form 9

Description: Defines the material properties for linear, temperature-independent, anisotropic materials for solid isoparametric elements

Format and Example:

1	2	3	4	5	6	7	8	9	10
MAT9	MID	G11	G12	G13	G14	G15	G16	G22	CONT
CONT	G23	G24	G25	G26	G33	G34	G35	G36	CONT
CONT	G44	G45	G46	G55	G56	G66	RHO	A1	CONT
CONT	A2	A3	A4	A5	A6	TREF	GE		

MAT9	17	6.2+3						6.2+3	ABC
+BC									DEF
+EF	5.1+3			5.1+3		5.1+3	3.2	6.6-6	GHI
+HI	6.5-6	6.5-6				125.			

Field	Contents
MID	Material identification number (Integer > 0)
G _{ij}	Elements of the 6x6 symmetric material property matrix (Real ≥ 0.0)
RHO	Mass density (Real ≥ 0.0)
A _i	Thermal expansion coefficient vector (Real)
TREF	Thermal expansion reference temperature (Real)
GE	Structural element damping coefficient (Real)

Remarks:

1. The material identification numbers must be unique for all **MAT1**, **MAT2**, **MAT8**, and **MAT9** entries.
2. The mass density **RHO** will be used to automatically compute mass in a structural dynamics problem.
3. Continuation number 4 need not be used.
4. The subscripts 1 through 6 refer to x, y, z, xy, yz, zx, for example:

$$\begin{Bmatrix} \sigma_x \\ \sigma_y \\ \sigma_z \\ \tau_{xy} \\ \tau_{yz} \\ \tau_{zx} \end{Bmatrix} = \begin{Bmatrix} G_{11} & & & & & \\ G_{12} & G_{22} & & & & \\ G_{13} & G_{23} & G_{33} & & & \\ G_{14} & G_{24} & G_{34} & G_{44} & & \\ G_{15} & G_{25} & G_{35} & G_{45} & G_{55} & \\ G_{16} & G_{26} & G_{36} & G_{46} & G_{56} & G_{66} \end{Bmatrix} \begin{Bmatrix} \epsilon_x \\ \epsilon_y \\ \epsilon_z \\ \gamma_{xy} \\ \gamma_{yz} \\ \gamma_{zx} \end{Bmatrix} - \begin{Bmatrix} A_1 \\ A_2 \\ A_3 \\ A_4 \\ A_5 \\ A_6 \end{Bmatrix} (T - T_0)$$

5. The damping coefficient, **GE** is:

$$GE = 2 \frac{C}{C_0}$$

Input Data Entry: **MFORM** Mass Matrix Form

Description: Defines the form of the mass matrix as consistent (coupled) or lumped.

Format and Example:

1	2	3	4	6	5	7	8	9	10
MFORM	VALUE								
MFORM	LUMPED								

Field	Contents
VALUE	A character string denoting the form of the mass matrix. The available forms are: 1) LUMPED 2) COUPLED

Remarks:

1. If more than one **MFORM** is included in the Bulk Data, any **COUPLED** value will result in coupled mass being used.
2. If no **MFORM** is indicated, the **LUMPED** formulation will be used.

Input Data Entry: MKAERO1 Mach Number - Frequency Table

Description: Provides a table of Mach numbers (**m**) and reduced frequencies (**k**) for unsteady aerodynamic matrix calculation.

Format and Example:

1	2	3	4	5	6	7	8	9	10
MKAERO1	SYMXZ	SYMXY	m ₁	m ₂	m ₃	m ₄	m ₅	m ₆	CONT
CONT	k ₁	k ₂	k ₃	k ₄	k ₅	k ₆	k ₇	k ₈	
MKAERO1	1	0	0.1	0.7					+ABC
+ABC	.3	.6	1.0						

Field	Contents
SYMXZ, SYMXY	Symmetry flags (Integer). See Remarks 5 and 6.
m _i	List of from 1 to 6 Mach numbers (Real ≥ 0.0 or blank)
k _i	List of from 1 to 8 reduced frequencies (Real ≥ 0.0 or blank)

Remarks:

1. All combinations of (**m**, **k**) will be used.
2. The continuation entry is required.
3. Several **MKAEROi** entries may be in the input packet. If these data entries are in the packet, they will be used.
4. The symmetry flags have the following definition:
 - +1 for symmetric
 - 0 for asymmetric
 - 1 for antisymmetric

The **m-k** pairs generated by this entry will generate aerodynamic matrices having the symmetries selected.

5. **m-k** pairs may be repeated with different symmetry options.
6. These are the following restrictions associated with the symmetry flags:
 - a) Ground effect is limited to antisymmetric only, **SYMXY** = 0 or -1.
 - b) Ground effect is not available at all for supersonic flow.
7. Reduced frequency is computed using:

$$k = \frac{b \omega}{2v}$$

where **b** is the reference chord defined by an **AERO** entry, ω is the frequency in radians per sec, and **v** is the true velocity.

Input Data Entry: MKAERO2 Mach Number - Frequency Table

Description: Provides a list of Mach numbers (**m**) and reduced frequencies (**k**) for aerodynamic matrix calculation.

Format and Example:

1	2	3	4	5	6	7	8	9	10
MKAERO2	SYMXZ	SYMXY	m ₁	k ₁	m ₂	k ₂	m ₃	k ₃	CONT
CONT	m ₄	k ₄	m ₅	k ₅	-etc-				
MKAERO2	0	0	.10	.60	.70	.30	.70	1.0	ABC
+BC	.8	.9	.8	1.0					

Field	Contents
SYMYZ, SYMXY	Symmetry flags (Integer). See Remarks 4 and 5.
m _i , k _i	List of pairs of Mach numbers (Real ≥ 0.) and reduced frequencies (real ≥ 0.)

Remarks:

1. This entry will cause the aerodynamic matrices to be computed for the given sets of parameter pairs.
2. Several **MKAERO2** entries may be in the input packet. If these data entries are in the packet, they will be used.
3. Any number of continuations are allowed.
4. The symmetry flags have the following definition:
 - +1 for symmetric (not available for **SYMXY**)
 - 0 for asymmetric
 - 1 for antisymmetric

The **m-k** pairs listed on the entry will generate aerodynamic matrices having the symmetries selected.

5. **m-k** pairs may be repeated with different symmetry options.
6. The following restrictions are imposed on the symmetry flags:
 - a) Ground effect (if present) must be antisymmetric **SYMXY** = 0 or -1.
 - b) Ground effect is not available at all for supersonic flow.
7. Reduced frequency is computed using:

$$k = \frac{b\omega}{2v}$$

where **b** is the reference chord defined by an **AERO** entry, ω is the frequency in radians per sec, and **v** is the true velocity.

Input Data Entry: MODELIST

Description: Defines a list of modes at which outputs are desired.

Format and Example:

1	2	3	4	5	6	7	8	9	10
MODELIST	SID	MODE1	MODE2	MODE3	MODE4	MODE5	MODE6	MODE7	CONT
CONT	MODE8	MODE9	-etc-						CONT

MODELIST	100	1	2	4					
----------	-----	---	---	---	--	--	--	--	--

Alternate Form:

1	2	3	4	5	6	7	8	9	10
MODELIST	SID	MODE1	THRU	MODE2					

Field	Contents
-------	----------

SID Set identification number referenced by Solution Control (Integer > 0)

MODEi Mode number of mode at which outputs are desired. (Integer > 0)

Remarks:

1. In order to be used, the **SID** must be referenced by Solution Control.
2. If the alternate form is used **MODE2** must be greater than or equal to **MODE1**.
3. Modes are numbered from 1 to n, starting at the lowest frequency for which a eigenvector was computed.
4. Nonexistent modes may be referenced and will result in no error message.

Input Data Entry: **MOMENT** Static Moment

Description: Defines a static moment at a grid point by specifying a vector.

Format and Example:

1	2	3	4	5	6	7	8	9	10
MOMENT	SID	G	CID	M	N1	N2	N3		
MOMENT	2	5	6	2.9	0.0	1.0	0.0		

Field	Contents
SID	Load set identification number (Integer > 0)
G	Grid point identification number (Integer > 0)
CID	Coordinate system identification number (Integer ≥ 0)
M	Scale factor (Real)
Ni	Components of vector measured in coordinate system defined by CID (Real; at least one nonzero component)

Remarks:

1. The static moment applied to grid point **G** is given by
 $\{m\} = M \{N1, N2, N3\}$
2. A **CID** of zero references the basic coordinate system.

Input Data Entry: **MOMENT1** Static Moment, Alternate Form 1

Description: Defines a static moment by specification of a value and two grid points which determine the direction.

Format and Example:

1	2	3	4	5	6	7	8	9	10
MOMENT	SID	G	M	G1	G2				
MOMENT	6	13	-2.93	16	13				

Field	Contents
SID	Load set identification number (Integer > 0)
G	Grid point identification number (Integer > 0)
M	Value of moment (Real)
Gi	Grid point identification numbers (Integer > 0; G1 ≠ G2)

Remarks:

1. The direction of the moment vector is determined by the vector from **G1** and **G2**.

Input Data Entry: **MPC** Multipoint Constraint

Description: Defines a multipoint constraint equation of the form

$$\sum_j A_j u_j = 0.0$$

Format and Example:

1	2	3	4	5	6	7	8	9	10
MPC	SID	G0	C0	A0	G	C	A		CONT
CONT		G	C	A	G	C	A		
MPC	3	28	3	6.2	2	3	4.29		+B
+B		1	4	-2.91					

Field	Contents
SID	Set identification (Integer > 0)
G0, G	Identification number of grid or scalar point (Integer > 0)
C0, C	Component number - any one of the digits 1 through 6 in the case of geometric grid points; blank or zero in the case of scalar points (Integer)
A0, A	Coefficient (Real; A0 must be nonzero)

Remarks:

1. The first coordinate (G0, C0) in the sequence is assumed to be the dependent coordinate. A dependent degree of freedom assigned by one MPC entry cannot be assigned dependent by another MPC entry or by a rigid element.
2. Forces of multipoint constraint are not recovered.
3. Multipoint constraint sets must be selected in Solution Control (MPC = SID) to be used.
4. The m-set coordinates specified on this entry may not be specified on other entries that define mutually exclusive sets.

Input Data Entry: MPCADD Multipoint Constraint Set Combination

Description: Defines a multipoint constraint set as a union of multipoint constraint sets defined via **MPC** entries.

Format and Example:

1	2	3	4	5	6	7	8	9	10
MPCADD	SID	S1	S2	S3	S4	S5	S6	S7	CONT
CONT	S8	S9	-etc-						
MPCADD	101	2	3	1	6	4			

Field	Contents
SID	Set identification number (Integer > 0)
S _j	Set identification numbers of multipoint constraint sets defined via MPC entries (Integer > 0)

Remarks:

1. The **S_j** must be unique.
2. Multipoint constraint sets must be selected in Solution Control (**MPC = SID**) to be used.
3. **S_j** may not be the identification number of a multipoint constraint set defined by another **MPCADD** entry.
4. **MPCADD** entries take precedence over **MPC** entries. If both have the same set identification number, only the **MPCADD** entry will be used.

Input Data Entry: MPPARM

Description: Identify values of user defined optimizer parameters that overrides the default values.

Format and Example:

1	2	3	4	5	6	7	8	9	10
MPPARM	PARAM	VALUE	PARAM	VALUE	PARAM	VALUE	PARAM	VALUE	CONT
CONT	PARAM	VALUE	-etc-						
MPPARM	ISCAL	0	STOL	0.005					

Field	Contents
PARAM	Name of parameter to be overridden (Text)
VALUE	Integer or real value to be used for the parameter.

Remarks:

1. Any number of **PARAM-VALUE** combinations can be specified on an **MPPARM** entry.
2. See EDO software manual (ADS V 1.10) for a definition of parameters, but the most useful are shown below:

REAL PARAMETER	DEFINITION	DEFAULT
CT	Constraint tolerance in the Method of Feasible Directions or the Modified Method of Feasible Directions. A constraint is active if its numerical value is more positive than CT.	-0.003
CTL	Same as CT, but for linear constraints.	-0.003
CTLMIN	Same as CTMIN, but for linear constraints.	0.0005
CTMIN	Minimum constraint tolerance for nonlinear constraints. If a constraint is more positive than CTMIN, it is considered to be violated.	0.0005
DABOBJ	Maximum absolute change in the objective between two consecutive iterations to indicate convergence in optimization.	max(0.001, $ F_1 - F_2 $, 0.0001)
DABOEM	Absolute convergence criterion for the optimization sub-problem when using sequential minimization techniques.	(Note 3)
DABSTR	Same as DABOBJ, but used at the strategy level.	(Note 3)
DELOBJ	Maximum relative change in the objective between two consecutive iterations to indicate convergence in optimization.	0.001

REAL PARAMETER	DEFINITION	DEFAULT
DELOBM	Relative convergence criterion for the optimization sub-problem when using sequential minimization techniques.	(Note 3)
DELSTR	Same as DELOBJ , but used at the strategy level.	(Note 3)
DOBJ1	Relative change in the objective function attempted on the first optimization iteration. Used to estimate initial move in the one-dimensional search. Updated as the optimization progresses.	0.1
DOBJ2	Absolute change in the objective function attempted on the first optimization iteration. Used to estimate initial move in the one-dimensional search. Updated as the optimization progresses.	$0.2 \max(X_i)$
DX1	Maximum relative change in a design variable attempted on the first optimization iteration. Used to estimate initial move in the one-dimensional search. Updated as the optimization progresses.	0.01
DX2	Maximum absolute change in a design variable attempted on the first optimization iteration. Used to estimate initial move in the one-dimensional search. Updated as the optimization progresses.	0.02
EXTRAP	Maximum multiplier on the one-dimensional search parameter, ALPHA in the one-dimensional search using polynomial interpolation/extrapolation.	(Note 3)
SCFO	The user-simplified value of the scale factor for the objective function if the default or calculated value is to be overridden.	(Note 3)
SCLMIN	Maximum numerical value of any scale factor allowed.	(Note 3)
STOL	Tolerance on the components of the calculated search direction to indicate that the Kuhn-Tucker conditions are satisfied.	(Note 3)
THETAZ	nominal value of the push-off factor in the Method of Feasible Directions.	(Note 3)

REAL PARAMETER	DEFINITION	DEFAULT
XMULT	Multiplier on the move parameter, ALPHA , in the one-dimensional search to find bounds on the solution.	(Note 3)
ZRO	Numerical estimate of zero on the computer. Usually the default value is adequate. If a computer with a short word length is used, ZRO = 1.0E-4 may be preferred.	(Note 3)

INTEGER PARAMETER	DEFINITION	DEFAULT
ISCAL	Scaling parameter. By default, scaling is done every NDV iterations, otherwise scaling is performed every ISCA iterations.	-1
ITMAX	Maximum number of iterations allowed at the optimizer level.	40
ITROMP	The number of consecutive iterations for which the absolute or relative convergence criteria must be met to indicate convergence at the optimizer level.	2
ITRMST	The number of consecutive iterations for which the absolute or relative convergence criteria must be met to indicate convergence at the optimizer level.	(Note 3)
JTMAX	Maximum of iterations allowed at the strategy level.	(Note 3)

3. Some of these parameters, indicated in the tables, are used only with the original version of the ADS optimizer. They are not used in MicroDOT.

Input Data Entry OCPARM Optimality Criteria Parameters

Description: Identifies values of user defined optimizer parameters that override default parameters

Format and Example:

1	2	3	4	5	6	7	8	9	10
OCPARM	OCPARM	PARAM	VALUE	PARAM	VALUE	PARAM	VALUE	PARAM	CONT
CONT	VALUE	PARAM	VALUE	ETC					
OCPARM	OCPARM	ISCAL	1						

Field	Contents
PARAM	Name of parameter to be overridden (Text)
VALUE	Integer or real value to be used for the parameter

Remarks:

- Any number of **PARAM/VALUE** pairs may be specified
- Legal values of **PARAM** are shown in the table below:

REAL PARAMETERS				INTEGER PARAMETERS			
PARAM	Default	PARAM	Default	PARAM	Default	PARAM	Default
EPSRED	0.0001	EPSOC	0.01	IUNIT	Note 1	ITER	NONE
EPSBNI	0.01	EPSFIN	0.001	ISTAT	NONE	NLPP	Note 3
EPSBNF	0.03	EPSCTB	0.1	ALNCSC	3	IPRINT	0
EPSCON	0.1	EPSGLB	0.9	NCSC	NONE	NRTAIN	NONE
XPAND	2.0	REDUC	0.5	NCBRZ	1	IGRDST	NONE
THRESH	0.05	OBJRSZ	NONE	ALSCMN	3	HSCALE	1
EPSOBJ	0.005			MXRTAN	Note 2	MXITR8	30
1. ASTROS Output Unit							
2. Minimum of 100 and the Number of Constraints							
3. ASTROS Number of Lines per Page							

Input Data Entry: **OMIT** Omitted Coordinates

Description: Defines degrees of freedom that the user desires to omit from the problem through matrix partitioning. Used to reduce the number of independent degrees of freedom.

Format and Example:

1	2	3	4	5	6	7	8	9	10
OMIT	SETID	ID	C	ID	C	ID	C		
OMIT	10	16	2	23	3516	54	23		

Field	Contents
SETID	The reduce set identification number (Integer > 0).
ID	Grid or scalar point identification number (Integer > 0).
C	Component number, zero or blank for scalar points, any unique combination of the digits 1 through 6 for grid points.

Remarks:

1. Coordinates specified on this entry form members of a mutually exclusive set. They may not be specified on other entries that define mutually exclusive sets.
2. In many cases it may be more convenient to use **OMIT1**, **ASET** or **ASET1** entries.

Input Data Entry: OMIT1 Omitted Coordinates, Alternate Form

Description: Defines degrees of freedom that the user desires to omit from the problem through matrix partitioning. Used to reduce the number of independent degrees of freedom.

Format and Example:

1	2	3	4	5	6	7	8	9	10
OMIT1	SETID	C	GID1	GID2	GID3	GID4	GID5	GID6	CONT
CONT	GID7	GID8	-etc-						

OMIT1	3	2	1	3	10	9	6	5	ABC
+BC	7	8							

Alternate Form:

1	2	3	4	5	6	7	8	9	10
OMIT1	SETID	C	GID1	THRU	GID2				

Field	Contents
SETID	The reduce set identification number (Integer > 0).
C	Component number (Any unique combination of the digits 1 through 6 (with no embedded blanks) when point identification numbers are grid points; must be null or zero if point identification numbers are scalar points).
GID _i	Grid or scalar point identification number (Integer > 0).

Remarks:

1. Coordinates specified on this entry form members of a mutually exclusive set. They may not be specified on other entries that define mutually exclusive sets.
2. If the alternate form is used, points in the sequence ID1 through ID2 are required to exist and ID2 must be greater than or equal to ID1.

Input Data Entry: **PAERO1** Aerodynamic Panel Property

Description: Gives associated bodies for the panels in the unsteady aerodynamic model.

Format and Examples:

1	2	3	4	5	6	7	8	9	10
PAERO1	PID	B1	B2	B3	B4	B5	B6		
PAERO1	1	3							

Field	Contents
PID	Property identification number (referenced by CAERO1) (Integer > 0)
Bi	Identification number of CAERO2 entries for associated bodies (Integer ≥ 0, or blank)

Remarks:

1. The associated bodies must be in the same aerodynamic group.
2. The **Bi** numbers above must appear on a **CAERO2** entry to define these bodies completely.

Input Data Entry: **PAERO2** Aerodynamic Body Properties

Description: Defines the cross-section properties of unsteady aerodynamic bodies.

Format and Examples:

1	2	3	4	5	6	7	8	9	10
PAERO2	PID	ORIENT	WIDTH	AR	LRSB	LRIB	LTH1	LTH2	CONT
CONT	THI1	THN1	THI2	THN2	THI3	THN3			

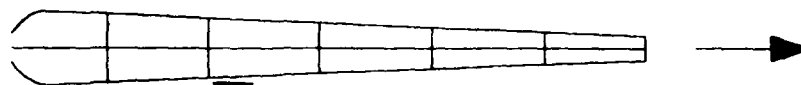
PAERO2	2	Z	6.0	1.0	22	91	100		abc
+bc	1	3							

Field	Contents
PID	Property identification number (Integer > 0)
ORIENT	Orientation flag "Z", "Y", or "ZY". Type of motion allowed for bodies (Text). Refers to the aerodynamic coordinate system "y" and "z" directions (see AERO data entry)
WIDTH	Reference half-width of body (Real > 0.0)
AR	Aspect ratio (height/width) (Real > 0.0)
LRSB	Identification number of an AEFACT data entry containing a list of slender body half-widths. If blank, the value of WIDTH will be used (Integer > 0 or blank)
LRIB	Identification number of an AEFACT data entry containing a list of interference body half-widths. If blank, the value of WIDTH will be used (Integer > 0 or blank)
LTH1, LTH2	Identification number of AEFACT data entries for defining theta arrays for interference calculations (Integer ≥ 0)
THIi, THNi	The first and last interference element of a body to use the θ_i array (Integer ≥ 0)

Remarks:

1. The **EID** of all **CAERO2** elements in any **IGID** group must be ordered, so that their corresponding **ORIENT** values appear in the order **Z**, **ZY**, **Y**.
2. The half-widths (given on **AEFACT** data entries referenced in fields 6 and 7) are specified at division points. The number of entries on an **AEFACT** data entry used to specify half-widths must be one greater than the number of elements.
3. The half-width at the first point (i.e., the nose) on a slender body is usually 0.; thus, it is recommended (but not required) that the **LRSB** data is supplied with a zero first entry.
4. **THIi** and **THNi** are interference element locations on a body. The element numbering begins at one for each body.
5. A body is represented by a slender body surrounded by an interference body. The slender body creates the down wash due to the motion of the body, while the interference body represents the effects upon panels and other bodies.

SLENDER BODY
(Six Elements Shown)

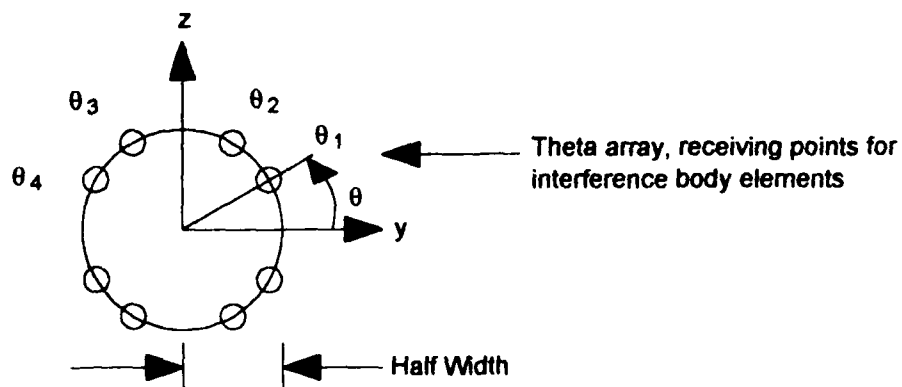


Division Points

INTERFERENCE BODY
(Three Elements Shown)



END VIEW
(Looking Forward)



Input Data Entry: PAERO6

Description: Defines body analysis parameters for steady aerodynamics.

Format and Examples:

1 2 3 4 5 6 7 8 9 10

PAERO6	BCID	CMPNT	CP	IGRP	NRAD	LRAD	LAXIAL		
PAERO6	10	FUSEL	0	3	4				

Field	Contents
BCID	Body component identification number (Integer > 0)
CMPNT	Component type (FUSEL for the fuselage and POD for a POD)
CP	Coordinate system of the geometry input (Integer ≥ 0, or blank)
IGRP	Group flag (Integer > 0)
NRAD	Number of equal radial cuts used to define the body panels (Integer ≥ 0 or blank)
LRAD	Identification number of an AEFACT data entry which defines the angular locations in degrees of the body panels (Integer ≥ 0 or blank)
LAXIAL	Identification number of an AEFACT data entry which defines the axial locations of in degrees of the body panels (Integer ≥ 0 or blank)

Remarks:

1. **NRAD** and **LRAD** are mutually exclusive.
2. If **LRAD** and **NRAD** are zero or blank, the radial cuts specified by the **BODY** or **AXSTA** entries are used.
3. **LAXIAL** is used only for **FUSEL** components. Inputs on the **AEFACT** entry are the dimensional fuselage stations.
4. If **LAXIAL** is blank, the axial locations are the same as those given by **AXSTA** data entries for the given body component.

Input Data Entry: PBAR Simple Beam Property

Description: Defines the properties of a simple beam (bar) which is used to create bar elements via the **CBAR** entry.

Format and Examples:

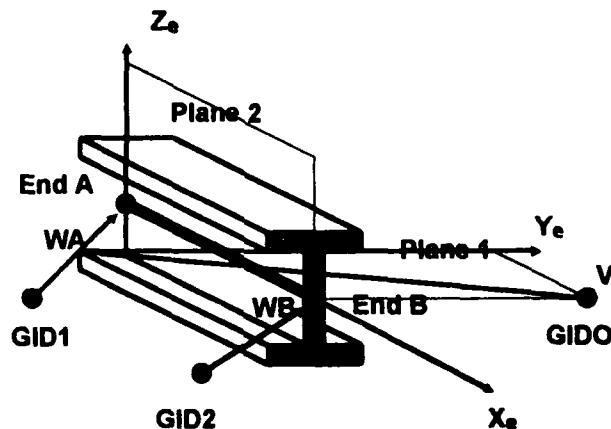
1	2	3	4	5	6	7	8	9	10
PBAR	PID	MID	A	I1	I2	J	NSM	TMIN	CONT
CONT	C1	C2	D1	D2	E1	E2	F1	F2	CONT
CONT	K1	K2	I12	R12	R22	ALPHA			

PBAR	39	6	2.9		5.97				123
+23			2.0	4.0					

Field	Contents
PID	Property identification number (Integer > 0)
MID	Material identification number (Integer > 0)
A	Area of bar cross-section (Real ≥ 0.0)
Ii	Area moments of inertia (Real) ($I1 \geq 0.0$, $I2 \geq 0.0$, $I1I2 > I12^2$)
J	Torsional constant (Real ≥ 0.0)
NSM	Nonstructural mass per unit length (Real ≥ 0.0)
TMIN	The minimum cross-sectional area in design (Real, Default = 0.0001)
K1, K2	Area factor for shear (Real)
Ci, Di, Ei, Fi	Stress recovery coefficients (Real)
R12, R22, ALPHA	Inertia linking terms for design (see Remarks 5 and 6)

Remarks:

1. The **BAR** element geometry and coordinate system is shown in the Figure on the following page.
2. **PBAR** entries may only reference **MAT1** material entries.
3. The transverse shear stiffnesses in planes 1 and 2 are $(K1)AG$ and $(K2)AG$, respectively. The default values for **K1** and **K2** are infinite. In other words, the transverse shear flexibilities are set equal to zero. **K1** and **K2** are ignored if $I12 \neq 0$.
4. The stress recovery coefficients **C1** and **C2**, etc., are the y and z coordinates in the **BAR** element coordinate system of a point at which stresses are computed. Stresses are computed at both ends of the **BAR**.
5. The **TMIN** value is used only for shape function design variable linking.



6. For design, the following applies to the **R12** and **R22** values. The moments of inertia are linked to the cross-sectional area by the following expressions:
- $$I1 = R12 * A^{**}ALPHA$$
- $$I2 = R22 * A^{**}ALPHA$$
- (A) If **R12** = 0.0 then the missing value is computed from $R12 = I1 / (A^{**}ALPHA)$. The same is true for **R22** and **I2**.
- (B) The **ALPHA** value defaults to 1.0 and must be ≥ 1.0 .
- (C) If both **I1** and **R12** or **I2** and **R22** are given, the linking expression will override the input I_i values.
7. If the **CBAR** is to be designed, the following restrictions apply.
- (A) $J = NSM = K1 = K2 = I12 = 0.0$
- If any of these values are not zero, a warning message will be issued and the value set to zero.

Input Data Entry: PCOMP Layered Composite Element Property

Description: Defines the properties of an n-ply composite material laminate.

Format and Examples:

1	2	3	4	5	6	7	8	9	10
PCOMP	PID	Z0	NSM	SBOND	F.T.	TMIN		LOPT	CONT
CONT	MID1	T1	TH1	SOUT1	MID2	T2	TH2	SOUT2	CONT
CONT	MID3	T3	TH3	SOUT3	-etc-				

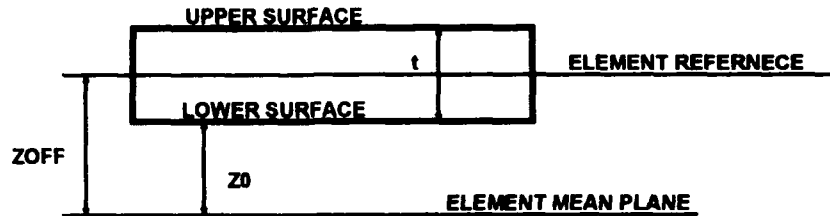
PCOMP	100	-0.5	1.5	5.+3	HOFF			MEM	ABC
+BC	150	0.05	90.	YES			-45.		DEF
+EF			45.0						

Field	Contents
PID	Property identification number (Integer > 0).
Z0	Offset of the laminate lower surface from the element mean plane. A positive value means the +z _e direction. (Real or blank, see Remark 2)
NSM	Nonstructural mass per unit area (Real ≥ 0.0).
SBOND	Allowable shear stress of the bonding material. (Real ≥ 0.0)
F.T.	Failure theory, one of the strings HILL , HOFF , TSAI , STRESS , or STRAIN . See Remark 4.
TMIN	Minimum ply thickness for design (Real > 0.0 or blank). (Default = 10 ⁻⁴)
LOPT	Lamination generation option, MEM or blank. See Remark 5.
MID _i	Material identification number of the i(th) layer. (Integer > 0 or blank)
T _i	Thickness of the i(th) layer (Real > 0.0 or blank).
TH _i	Angle between the longitudinal direction of the fibers of the i(th) layer and the material X-axis. (Real or blank)
SOUT _i	Stress output request for i(th) layer, one of the strings YES or NO . (Default = NO)

Remarks:

1. For non-designed elements, the plies are numbered from 1 to n beginning with the bottom layer.

2. For composites there are two methods for specifying the offset of the element reference plane from the element mean plane: **Z0** on this entry and **ZOFF** on the **CQUAD4** or **CTRIA3** Bulk Data entries. The distinction is shown in the figure below:



You may only specify a **Z0** on this entry if the **ZOFF** field of any **CQUAD4** or **CTRIA3** referencing it is blank. The default value for **Z0** is $-t/2$ where t is the overall thickness of the laminate.

3. **SBOND** is required if bonding material failure index calculations are desired.
4. The failure theory is used to determine the element failure on a ply-by-ply basis. The available theories are:
 - HILL** - Hill Theory
 - HOFF** - Hoffman Theory
 - TSAI** - Tsai-Wu Theory
 - STRESS**- For Maximum Stress Theory
 - STRAIN**- For Maximum Strain Theory
5. **MEM** indicates a layup of membrane only plies.
6. The material properties, **MID_i**, may reference only **MAT1**, **MAT2**, and **MAT8** Bulk Data entries.
7. If any of the **MID_i**, **T_i** or **TH_i** are blank, then the last non-blank values specified for each will be used to define the values for the ply.
8. **TMIN** will be ignored unless the element is linked to design variables by **SHAPE** entries.

Input Data Entry: **PCOMP1** Layered Composite Element Property

Description: Defines the properties of an n-ply laminated composite material where all plies are composed of the same material and are of equal thickness.

Format and Examples:

1	2	3	4	5	6	7	8	9	10
PCOMP1	PID	Z0	NSM	SBOND	F.T.	TMIN	MID	LOPT	CONT
CONT	TPLY	TH1	TH2	TH3	TH4	TH5	TH6	TH7	CONT
CONT	TH8	TH9	TH10	-etc-					

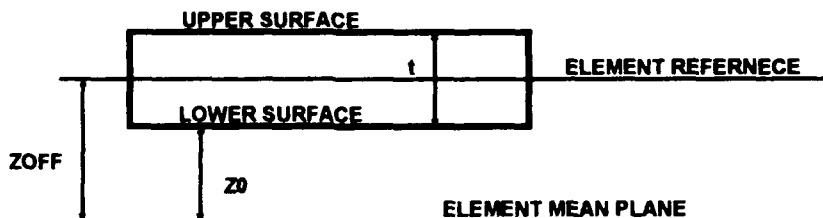
PCOMP1	100	-0.5	1.7	5.+3	STRAIN		200		ABC
+BC	0.25	-45.0	45.0	90.0	90.0	45.0			

Field	Contents
PID	Property identification number (1,000,000 > Integer > 0).
Z0	Offset of the laminate lower surface from the element mean plane. A positive value means the +z _e direction. (Real or blank, see Remark 2)
NSM	Nonstructural mass per unit area (Real ≥ 0.0).
SBOND	Allowable shear stress of the bonding material. (Real ≥ 0.0)
F.T.	Failure theory, one of the strings HILL , HOFF , TSAI , STRESS or STRAIN . See Remark 4.
TMIN	Minimum ply thickness for design (Real > 0.0 or blank) (Default = 0.0001)
MID	Material identification number for all layers. (Integer > 0.0 or blank)
LOPT	Lamination generation option, MEM or blank. See Remark 5.
TPLY	Thickness of each layer. (Real > 0.0).
TH _i	Angle between the longitudinal direction of the fibers of the i(th) layer and the material X-axis. (Real or blank)

Remarks:

1. For nondesigned elements, the plies are numbered from 1 to n beginning with the bottom layer.

- For composites there are two methods for specifying the offset of the element reference plane from the element mean plane: **Z0** on this entry and **ZOFF** on the **CQUAD4** or **CTRIA3** Bulk Data entries. The distinction is shown in the figure below:



You may only specify a **Z0** on this entry if the **ZOFF** field of any **CQUAD4** or **CTRIA3** referencing it is blank. The default value for **Z0** is $-t/2$ where t is the overall thickness of the laminate.

- SBOND** is required if bonding material failure index calculations are desired.
- The failure theory is used to determine the element failure on a ply-by-ply basis. The available theories are:
 - HILL** - Hill Theory
 - HOFF** - Hoffman Theory
 - TSAI** - Tsai-Wu Theory
 - STRESS**- For Maximum Stress Theory
 - STRAIN**- For Maximum Strain Theory
- MEM** indicates a layup of membrane only plies.
- The material properties, **MIDi**, may reference only **MAT1**, **MAT2**, and **MAT8** Bulk Data entries.
- TMIN** will be ignored unless the element is linked to design variables by **SHAPE** entries.

Input Data Entry: PCOMP2 Layered Composite Element Property

Description: Defines the properties of an n-ply laminated composite material where all plies are composed of the same material but are of different thickness.

Format and Examples:

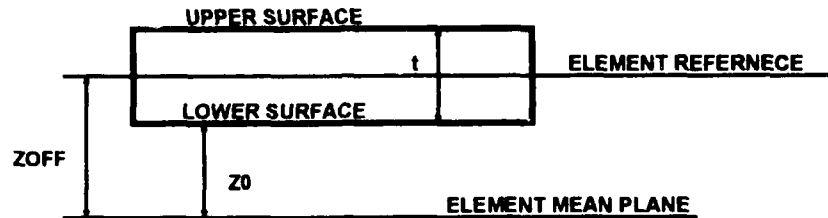
1	2	3	4	5	6	7	8	9	10
PCOMP2	PID	Z0	NSM	SBOND	F.T.	TMIN	MID	LOPT	CONT
CONT	T1	TH1	T2	TH2	T3	TH3	-etc-		
PCOMP2	100	-0.5	1.7	5.+3	TSAI		200		ABC
+BC	0.25	-45.0	0.5	90.0	0.25	45.0			

Field	Contents
PID	Property identification number (Integer > 0).
Z0	Offset of the laminate lower surface from the element mean plane. A positive value means the +z _e direction. (Real or blank, see Remark 2)
NSM	Nonstructural mass per unit area (Real ≥ 0.0).
SBOND	Allowable shear stress of the bonding material. (Real ≥ 0.0)
F.T.	Failure theory, one of the strings HILL , HOFF , TSAI , STRESS , or STRAIN . See Remark 4.
TMIN	Minimum ply thickness for design (Real > 0.0 or blank) (Default = 10 ⁻⁴)
MID	Material identification number for all layers. (Integer > 0.0 or blank)
LOPT	Lamination generation option, MEM or blank. See Remark 5.
T _i	Thickness of all layers. (Real > 0.0 or blank).
TH _i	Angle between the longitudinal direction of the fibers of the i(th) layer and the material X-axis. (Real or blank)

Remarks:

1. For nondesigned elements, the plies are numbered from 1 to n beginning with the bottom layer.

2. For composites there are two methods for specifying the offset of the element reference plane from the element mean plane: **Z0** on this entry and **ZOFF** on the **CQUAD4** or **CTRIA3** Bulk Data entries. The distinction is shown in the figure below:



You may only specify a **Z0** on this entry if the **ZOFF** field of any **CQUAD4** or **CTRIA3** referencing it is blank. The default value for **Z0** is $-t/2$ where t is the overall thickness of the laminate.

3. **SBOND** is required if bonding material failure index calculations are desired.
4. The failure theory is used to determine the element failure on a ply-by-ply basis. The available theories are:
 - HILL** - Hill Theory
 - HOFF** - Hoffman Theory
 - TSAI** - Tsai-Wu Theory
 - STRESS** - For Maximum Stress Theory
 - STRAIN** - For Maximum Strain Theory
5. **MEM** indicates a layup of membrane only plies.
6. The material properties, **MID_i**, may reference only **MAT1**, **MAT2**, and **MAT8** Bulk Data entries.
7. If any of the **T_i** or **TH_i** are blank, then the last non-blank values specified for each will be used to define the values for the ply.
8. **TMIN** will be ignored unless the element is linked to design variables by **SHAPE** entries.

Input Data Entry: **PELAS** Scalar Elastic Property

Description: Used to define the stiffness, damping coefficient, and stress coefficient of a scalar elastic element (spring) defined by means of the **CELAS1** entry.

Format and Example:

1	2	3	4	5	6	7	8	9	10
PELAS	PID	K	GE	S	TMIN				
PELAS	7	4.29	0.06	7.92					

Field	Contents
PID	Property identification number (Integer > 0)
K	Elastic property value (Real)
GE	Damping coefficient (Real \geq 0.0)
S	Stress coefficient (Real)
TMIN	Minimum value for design (Real > 0.0, or blank, Default = 0.0001)

Remarks:

1. The user is cautioned to be careful using negative spring values.
2. **TMIN** is ignored unless the element is designed using shape function design variable linking.

Input Data Entry: PIHEx Isoparametric Hexahedron Property

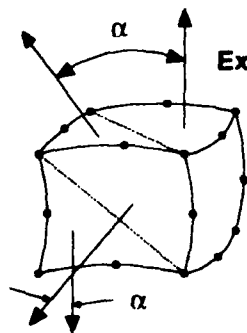
Description: Defines the properties of an isoparametric solid element, including a material reference and the number of integration points. Referenced by the **CIHEx1**, **CIHEx2**, and **CIHEx3** entries.

Format and Examples:

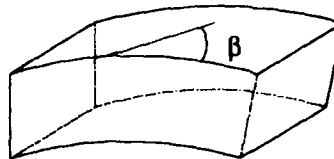
1	2	3	4	5	6	7	8	9	10
PIHEx	PID	MID	CID	NIP	AR	ALFA	BETA		
PIHEx	15	3		3			5.0		

Field	Contents
PID	Property identification number (Integer > 0)
MID	Material identification number (Integer > 0)
CID	Identification number of the coordinate system in which the material referenced by MID is defined (Integer ≥ or blank)
NIP	Number of integration points along each edge of the element (Integer = 2, 3, 4, or blank)
AR	Maximum aspect ratio (ratio of longest to shortest edge) of the element (Real > 1.0 or blank)
ALFA	Maximum angle in degrees between the normals of two subtriangles comprising a quadrilateral face (Real, $0.0 \leq \text{ALFA} \leq 180.0$ or blank) (Default = 45.0)
BETA	Maximum angle in degrees between the vector connecting a corner point to an adjacent midside point and the vector connecting that midside point and the other midside or corner point (Real, $0.0 < \text{BETA} < 180.0$ or blank) (Default = 45.0)

Examples of Field Definitions:



Example of ALPHA



Example of BETA

Remarks:

1. All **PIHEX** cards must have unique identification numbers.
2. **CID** is not used for isotropic materials.
3. The default for **CID** is the basic coordinate system.
4. The default for **NIP** is 2 for **IHEX** and 3 for **IHEX2** and **IHEX3**.
5. **AR**, **ALFA**, and **BETA** are used for checking the geometry of the element. The defaults are:

	AR	ALFA (degrees)	BETA (degrees)
CIHEX1	5.0	45.0	--
CIHEX2	10.0	45.0	45.0
CIHEX3	15.0	45.0	45.0

Input Data Entry: **PLIST**

Description: Defines property entries associated with a design variable.

Format and Examples:

1	2	3	4	5	6	7	8	9	10
PLIST	LINKID	PTYPE	PID1	PID2	PID3	PID4	PID5	PID6	CONT
CONT	PID7	PID8	PID9	-etc-					

PLIST	6	PROD	12	14	22				
-------	---	------	----	----	----	--	--	--	--

Alternate Form:

1	2	3	4	5	6	7	8	9	10
PLIST	DVID	PTYPE	PID1	THRU	PID2				

Field	Contents
LINKID	Property list identifier (Integer > 0).
PTYPE	Property type associated with this list (e.g., PROD).
PID1, PID2, PID3	Property entry identifications. (Integer > 0, or blank)

Remarks:

1. Allowable **PTYPES** are: **PROD**, **PSHEAR**, **PCOMP**, **PCOMP1**, **PCOMP2**, **PELAS**, **PSHELL** , **PMASS**, **PTRMEM**, **PQDMEM1**, and **PBAR**.
2. If the alternate form is used, **PID2** must be greater than or equal to **PID1**.
3. All elements using properties listed on **PLIST** entries for a particular **LINKID** will be designed by (linked to) that design variable that references the **PLIST LINKID**.

Input Data Entry: **PLOAD** Static Pressure Load

Description: Defines a static pressure load on a triangular or quadrilateral surface.

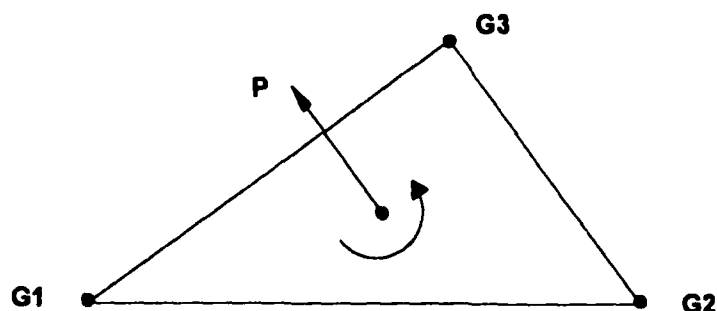
Format and Examples:

1	2	3	4	5	6	7	8	9	10
PLOAD	SID	P	G1	G2	G3	G4			
PLOAD	1	-4.0	16	32	11				

Field	Contents
SID	Load set identification number (Integer > 0)
P	Pressure (Real)
Gi	Grid point identification numbers (Integer > 0; G4 may be zero)

Remarks:

1. The grid points define either a triangular or a quadrilateral surface to which a pressure is applied. If G4 is zero or blank, the surface is triangular.
2. In the case of a triangular surface, the assumed direction of the pressure is computed according to the right-hand rule using the sequence of grid points G1, G2, and G3 as illustrated below.



The total load on the surface, AP, is divided into three equal parts and applied to the grid points as concentrated loads. A minus sign in field 3 reverses the direction of the load.

3. In the case of a quadrilateral surface, the grid points G1, G2, G3, and G4 should form a consecutive sequence around the perimeter. The right-hand rule is applied to find the assumed direction of the pressure. Four concentrated loads are applied to the grid points in approximately the same manner as for a triangular surface. The following specific procedures are adopted to accommodate irregular and/or warped surfaces:
 - a. The surface is divided into two sets of overlapping triangular surfaces. Each triangular surface is bounded by two of the sides and one of the diagonals of the quadrilateral.
 - b. One-half of the pressure is applied to each triangle which is then treated in the manner described in Remark 2.
4. Load sets must be selected in Solution Control to be used.

Input Data Entry: **PLYLIST** A list of composite element layer numbers.

Description: Defines a set of layers of composite elements by a list.

Format and Examples:

1	2	3	4	5	6	7	8	9	10
PLYLIST	SID	P1	P2	P3	P4	P5	P6	P7	CONT
CONT	P8	-etc-							

PLYLIST	3	1	2	3	4	16	15	14	ABC
+BC	13								

Alternate Form:

1	2	3	4	5	6	7	8	9	10
PLYLIST	SID	P1	THRU	P2					

Field

Contents

SID Set of identification numbers (Integer > 0)

Pi List of ply numbers (Integer > 0)

Remarks:

1. These entries are referenced by the **DESVARP**, **DESVARs**, **DCONLMN**, **DCONPMN**, **DCONLAM** and **DCONTH2** data entries.
2. When using the **THRU** option, all intermediate plies will be assumed to exist.
3. When used by **DESVARs** and **DESVARP**, the entry refers to composite layer numbers to be linked together in the design model.
4. When used by **DCONLMN**, **DCONPMN** and **DCONLAM**, the entry refers to composite layers that, together, define a "ply" or a "laminare" whose summed thicknesses will be contribute to the constraint.

Input Data Entry: **PMASS** Scalar Mass Property

Description: Used to define the mass value of a scalar mass element which is defined by means of the **CMASS1** entries.

Format and Examples:

1	2	3	4	5	6	7	8	9	10
PMASS	PID	M	TMIN	PID	M	TMIN			
PMASS	7	4.29	0.2	6	13.2	0.1			

Field	Contents
PID	Property identification number (Integer > 0).
M	Value of scalar mass (Real).
TMIN	The minimum mass value in design. Default = 0.0001

Remarks:

1. This entry defines a mass value.
2. Up to 2 mass values may be defined by this entry.
3. **TMIN** is ignored unless the mass element is linked to design variables through **SHAPE** entries.

Input Data Entry: **PQDMEM1** **Quadrilateral Membrane Property**

Description: Used to define the properties of a quadrilateral membrane referenced by the **COMMEM1** entry. No bending properties are included.

Format and Examples:

1	2	3	4	5	6	7	8	9	10
PQDMEM1	PID	MID	T	NSM	TMIN				
PQDMEM1	235	2	0.5	0.0					

Field	Contents
PID	Property identification number (Integer > 0).
MID	Material identification number (Integer > 0).
T	Thickness of membrane (Real \geq 0.0)
NSM	Nonstructural mass per unit area (Real \geq 0.0).
TMIN	Minimum thickness for design (Real > 0.0 or blank) (Default = 0.0001)

Remarks:

1. All **PQDMEM1** entries must have unique property identification numbers.
2. **TMIN** is ignored unless the element is linked to the global design variables by a **SHAPE** entry.

Input Data Entry: **PROD** Rod Property

Description: Defines the properties of a rod which is referenced by the **CROD** entry.

Format and Examples:

1	2	3	4	5	6	7	8	9	10
PROD	PID	MID	A	J	C	NSM	TMIN		
PROD	17	23	42.6	17.92	4.236	0.5			

Field	Contents
PID	Property identification number (Integer > 0)
MID	Material identification number (Integer > 0)
A	Area of rod (Real ≥ 0.0, or blank)
J	Torsional constant (Real ≥ 0.0, or blank)
C	Coefficient to determine torsional stress (Real ≥ 0.0, or blank)
NSM	Nonstructural mass per unit length (Real ≥ 0.0, or blank)
TMIN	Minimum rod area for design (Real > 0.0, or blank). Default = 0.0001

Remarks:

1. **PROD** entries must all have unique property identification numbers.
2. For structural problems, **PROD** entries may only reference **MAT1** material entries.
3. The formula used to compute torsional stress is:

$$\tau = \frac{CM_\theta}{J}$$

where M_θ is the torsional moment.
4. **TMIN** is ignored unless the rod element is linked to the design variables by **SHAPE** entries.

Input Data Entry: PSHEAR Shear Panel Property

Description: Defines the elastic properties of a shear panel. Referenced by the **CSHEAR** entry.

Format and Examples:

1	2	3	4	5	6	7	8	9	10
PSHEAR	PID	MID	T	NSM	TMIN				
PSHEAR	13	2	4.9	16.2					

Field	Contents
PID	Property identification number (Integer > 0)
MID	Material identification number (Integer > 0)
T	Thickness of shear panel (Real > 0.0)
NSM	Nonstructural mass per unit area (Real ≥ 0.0, or blank)
TMIN	Minimum panel thickness for design (Real ≥ 0.0, or blank). Default = 0.0001

Remarks:

1. All **PSHEAR** entries must have unique identification numbers.
2. **PSHEAR** entries may reference only **MAT1** material entries.
3. **TMIN** is ignored unless the element is linked to global design variables by **SHAPE** entries.

Input Data Entry: PSHELL Shell Element Property

Description: Defines the membrane, bending, transverse shear, and coupling properties of the shell elements. (QUAD4 and TRIA3)

Format and Examples:

1	2	3	4	5	6	7	8	9	10
PSHELL	PID	MID1	T	MID2	12I/T3	MID3	TS/T	NSM	CONT
CONT	Z1	Z2	MID4	MCSID	SCSID	ZOFF	TMIN		
PSHELL	203	204	1.90	205	1.2	206	0.8	6.32	ABC
+BC	+.95	-.95		0	0	0.01			

Field	Contents
PID	Property identification number (Integer > 0)
MID1	Material identification number for membrane (Integer > 0 or blank)
T	Default value for membrane thickness (Real > 0.0, or blank)
MID2	Material identification number for bending (Integer > 0, or blank)
12I/T3	Bending stiffness parameter (Real > 0.0, or blank, Default = 1.0)
MID3	Material identification number for transverse shear (Integer > 0, or blank), must be blank unless MID2 > 0)
TS/T	Transverse shear thickness divided by membrane thickness (Real > 0.0 or blank, Default = .833333).
NSM	Nonstructural mass per unit area (Real > 0.0, or blank)
Z1, Z2	Fiber distances for stress computation. The positive direction is determined by the right-hand rule and the order in which the grid points are listed on the connection entry. (Real or blank, defaults are -1/2 T for Z1 and 1/2 T for Z2.)
MID4	Material identification number for membrane-bending coupling (Integer > 0 or blank, must be blank unless MID1 > 0 and MID2 > 0, may not equal MID1 or MID2)
MCSID	Identification number of material coordinate system (Real or blank, or Integer ≥ 0) (See Remark 9)
SCSID	Identification number of stress coordinate system (Real or blank, or Integer ≥ 0) (See Remark 9)
ZOFF	Offset of the element reference plane from the plane of grid points. A positive value means the +z _e direction. (Real or blank, default = 0.0) (See Remark 10)
TMIN	Minimum thickness for design (Real > 0.0 or blank) (Default = 0.0001)

Remarks:

1. All PSHELL property entries must have unique identification numbers.
2. The structural mass is computed from the density using the membrane material properties.

3. The results of leaving an **MID** field blank are:
 - MID1** No membrane or coupling stiffness.
 - MID2** No bending, coupling, or transverse shear stiffness.
 - MID3** No transverse shear flexibility.
 - MID4** No bending-membrane coupling.
4. The continuation entry is not required.
5. The **MID4** field should be left **blank** if the material properties are symmetric with respect to the middle surface of the shell.
6. This entry is used only with the **CQUAD4** elements.
7. For structural problems, **PSHELL** entries may reference **MAT1**, **MAT2**, or **MAT8** material property entries.
8. If the transverse shear material, **MID3**, references **MAT2** data, then **G33** must be $\neq 0$. If **MID3** references **MAT8** data, then **G1, Z** and **G2, Z** must not be zero.
9. If **MCSID/SCSID** is left blank (0.0) or is real, it is considered to be the angle of rotation of the **X** axis of the material/stress coordinate system with respect to the **X** axis of the element coordinate system in the **XY** plane of the latter. If Integer, the orientation of the material/stress **x**-axis is along the projection of the **x**-axis of the specified coordinate system onto the **x-y** plane of the element system. The value of **MCSID** is the default value for the **TM** field on **CQUAD4** Bulk Data entries.
10. The offset **ZOFF** may also be provided on the **CQUAD4** or **CTRIA3** Bulk Data entry. The element reference plane is located at the mid-thickness of the element parallel to the element mean plane.
11. **TMIN** is ignored unless element is linked to global design variables by **SHAPE** entries.
12. The hierarchy of local coordinate systems is:
 - MCSID** supplies the default value for the **TM** field on the element connectivity entry
 - TM** overrides **MCSID** if **TM** is not blank
 - SCSID** defaults to the material coordinate system if **SCSID** is blank

Input Data Entry: PTRMEM

Description: Defines property data for ~~TRMEM~~ element.

Format and Examples:

1	2	3	4	5	6	7	8	9	10
PTRMEM	PID	MID	T	NSM	TMIN				
PTRMEM	500	1000	0.15						

Field	Contents
PID	Property entry identification number (Integer > 0)
MID	Material property identification (Integer > 0)
T	Thickness of membrane element (Real > 0.0)
NSM	Nonstructural mass associated with the element (Real > 0.0, or blank)
TMIN	Minimum thickness for design (Real > 0.0 or blank) (Default = 0.0001)

Remarks:

1. The ~~PTRMEM~~ entry can reference either **MAT1**, **MAT2** or **MAT8** entries.
2. ~~TMIN~~ is ignored unless the element is linked to global design variables by **SHAPE** entries.

Input Data Entry: RANDPS Power Spectral Density Specification

Description: Defines load set power spectral density factors for use in Random Analysis having the frequency dependent form.

$$S_{jk}(F) = (x + iy) G(F)$$

Format and Examples:

1	2	3	4	5	6	7	8	9	10
RANDPS	SID	J	K	X	Y	TID			
RANDPS	5	3	7	2.0	2.5	4			

Field	Contents
SID	Random analysis set identification number (Integer > 0)
J	Subcase identification number of excited load set (Integer > 0)
K	Subcase identification number of applied load set (Integer > 0; K > J)
X, Y	Components of complex number (Real)
TID	Identification number of a TABRND entry which defines $G(F)$ (Integer ≥ 0)

Remarks:

1. If J = K, then Y must be 0.0.
2. For TID = 0, $G(F) = 1.0$.
3. Set identification numbers must be selected in Solution Control to be used.
4. **RANDPS** can only reference subcases included within a single loop (change in direct matrix input is not allowed).

Input Data Entry **RBAR** Rigid Bar

Description: Defines a Rigid Bar element with six degrees of freedom at each end.

Format and Example:

1	2	3	4	5	6	7	8	9	10
RBAR	SETID	EID	GA	GB	CNA	CNB	CMA	CMB	
RBAR	1001	5	1	2	234	123			

Field	Contents
SETID	Multipoint constraint set identification number specified in Solution Control. (Integer > 0)
EID	Rigid Bar element identification number. (Integer > 0)
GA, GB	Grid point identification numbers of connection points. (Integer > 0)
CNA, CNB	Independent degrees of freedom in the global coordinate system for the elements at grid point GA and GB . Indicated by any of the digits 1 through 6 with no embedded blanks. (Integer ≥ 0, or blank) (Remark 2)
CMA, CMB	Component numbers of dependent degrees of freedom in the global coordinate system assigned by the element at grid point GA and GB . Indicated by any of the digits 1 through 6 with no embedded blanks. (Integer > 0 or blank) (Remarks 3 and 4)

Remarks:

1. The **RBAR** entry is selected in the Solution Control with the **MPC=SETID** option of the **BOUNDARY** command. **THIS IS AN ENHANCEMENT TO THE NASTRAN METHOD, WHICH DOES NOT ALLOW RIGID CONNECTIONS TO BE CHANGED FOR DIFFERENT BOUNDARY CONDITIONS.**
2. The total number of components in **CNA** and **CNB** must be six; for example, **CNA=1236**, **CNB=34**. The components must jointly be capable of representing any general rigid body motion of the element.
3. If both **CMA** and **CMB** are zero or blank, all of the degrees of freedom not in **CNA** and **CNB** will be made dependent, i.e. they will be placed in the m-set.
4. The m-set degrees of freedom specified on this entry may not be specified on other entries that define mutually exclusive sets.
5. Rigid element identification numbers must be unique within each element type for each **MPC** set identification number.

Input Data Entry RBE1**Rigid Body Element, Form 1**

Description: Defines a rigid body connected to an arbitrary number of grid points.

Format and Example:

1	2	3	4	5	6	7	8	9	10
RBE1	SETID	EID	GN1	CN1	GN2	CN2	GN3	CN3	CONT
CONT		GN4	CN4	GN5	CN5	GN6	CN6		CONT
CONT	"UM"	GM1	CM1	GM2	CM2	GM3	CM3		CONT
CONT		GM4	CM4	GM5	CM5	-etc-			

RBE1	1001	11	1	2	2	134	3	5	AB
AB		4	2						CD
CD	UM	1	13	2	1	12	5		

Field	Contents
SETID	Multipoint constraint set identification number specified in Solution Control. (Integer > 0)
EID	Rigid body element identification number. (Integer > 0)
GNi	Grid point identification numbers at which independent degrees of freedom are assigned. (Integer > 0)
CNi	Component numbers of independent degrees of freedom in the global coordinate system at grid points GNi , indicated by any of the digits 1 through 6 with no embedded blanks. (Integer > 0) (Remark 2)
"UM"	Character string indicating the start of the list of dependent degrees of freedom.
GMj	Grid point identification numbers at which dependent degrees of freedom are assigned. (Integer > 0)
CMj	Component numbers of dependent degrees of freedom in the global coordinate system at grid points GMj , indicated by any of the digits 1 through 6 with no embedded blanks. (Integer > 0) (Remark 2)

Remarks:

1. The **RBE1** entry is selected in the Solution Control with the **MPC=SETID** option of the **BOUNDARY** command. **THIS IS AN ENHANCEMENT TO THE NASTRAN METHOD, WHICH DOES NOT ALLOW RIGID CONNECTIONS TO BE CHANGED FOR DIFFERENT BOUNDARY CONDITIONS.**
2. The total number of components in **CNi** must be six; for example, **CN1=123**, **CN2=3**, **CN3=2** and **CN4=3**. The components must jointly be capable of representing any general rigid body motion of the element. The m-set degrees of freedom specified on this entry may not be specified on other entries that define mutually exclusive sets.
3. A degree-of-freedom cannot be both independent and dependent for the same element. However, both independent and dependent components may exist at the same grid point.
4. Rigid element identification numbers must be unique within each element type for each **MPC** set identification number.

Input Data Entry RBE2 Rigid Body Element, Form 2

Description: Defines a body whose independent degrees of freedom are specified at a single grid point and whose dependent degrees of freedom are specified at an arbitrary number of grid points.

Format and Example:

1	2	3	4	5	6	7	8	9	10
RBE2	SETID	EID	GN	CM	GM1	GM2	GM3	GM4	CONT
CONT	GM5	GM6	GM7	GM8	-etc-				
RBE2	1001	9	8	12	10	12	14	15	AB
AB	16	20							

Field	Contents
SETID	Multipoint constraint set identification number specified in Solution Control. (Integer > 0)
EID	Rigid body element identification number. (Integer > 0)
GN	Grid point identification number at which all 6 independent degrees of freedom are assigned. (Integer > 0)
CM	Component numbers of dependent degrees of freedom in the global coordinate system assigned by the element at grid points GM1 , GM2 , etc. Indicated by any of the digits 1 through 6 with no embedded blanks. (Integer > 0 or blank)
GMi	Grid point identification number at which dependent degrees of freedom are assigned. (Integer > 0)

Remarks:

1. The **RBE2** entry is selected in the Solution Control with the **MPC=SETID** option of the **BOUNDARY** command. **THIS IS AN ENHANCEMENT TO THE NASTRAN METHOD, WHICH DOES NOT ALLOW RIGID CONNECTIONS TO BE CHANGED FOR DIFFERENT BOUNDARY CONDITIONS.**
2. The components indicated by **CM** are made dependent at all grid points **GMi**.
3. The m-set degrees of freedom specified on this entry may not be specified on other entries that define mutually exclusive sets.
4. Rigid element identification numbers must be unique within each element type for each **MPC** set identification number.

Input Data Entry RBE3**Rigid Body Element, Form 3**

Description: Defines the motion of a reference grid point as the weighted average of motions at a set of other grid points.

Format and Example:

1	2	3	4	5	6	7	8	9	10
RBE3	SETID	EID	REFG	REFC	WT1	C1	G1,1	G1,2	CONT
CONT		G1,3	WT2	C2	G2,1	G2,2	-etc-	WT3	CONT
CONT		C3	G3,1	-etc-	-etc-	WT4	C4	G4,1	CONT
CONT		G4,2	-etc-						CONT
CONT	"UM"	GM1	CM1	GM2	CM2	GM3	CM3		CONT
CONT		GM4	CM4	-etc-					

RBE3	1001	14	100	1234	1.0	123	1	3	AE
+E		5	4.7	1	2	4	6	5.2	AF
+F		2	7	8	9	5.1	1	15	AG
+G		16							AH
+H	UM	100	14	5	3	7	2		

Field	Contents
SETID	Multipoint constraint set identification number specified in Solution Control. (Integer > 0)
EID	Rigid body element identification number. (Integer > 0)
REFG	Reference grid point identification number. (Integer > 0)
REFC	Component numbers of degrees of freedom in the global coordinate system that will be computed at REFG , Indicated by any of the digits 1 through 6 with no embedded blanks. (Integer > 0)
WTi	Weighting factor for most common defined by Gi,j . (Real)
Ci	Component numbers of degrees of freedom in the global coordinate system which have weighting factor WTi , at grid points Gi,j . Indicated by any of the digits 1 through 6 with no embedded blanks. (Integer > 0)
Gi,j	Grid point identification number whose components Ci have weighting factor WTi . (Integer > 0)
"UM"	Character string indicating the start of the list of dependent degrees of freedom. The default is that all of the components in REFC at REFG , and no others, will be placed in the m-set.
GMi	Grid point identification numbers with components in the m-set. (Integer > 0)
CMi	Component numbers in the global coordinate system at grid points GMi which are placed in the m-set. Indicated by any of the digits 1 through 6 with no embedded blanks. (Integer > 0 or blank) (Remark 2)

Remarks:

1. The **RBE3** entry is selected in the Solution Control with the **MPC=SETID** option of the **BOUNDARY** command. **THIS IS AN ENHANCEMENT TO THE NASTRAN METHOD, WHICH DOES NOT ALLOW RIGID CONNECTIONS TO BE CHANGED FOR DIFFERENT BOUNDARY CONDITIONS.**
2. The form of **G_{i,j}** is different than NASTRAN. The first data field on the continuations has been reserved for the "**UM**" identifier. The **G_{i,j}** list must be contained within data fields 3 through 9. Blanks may appear anywhere in the list.
3. The default for "**UM**" should be used except in cases where the user wishes to include some or all of the **REFC** components in displacement sets other than the m-set. If the default is not used for "**UM**" then:
the total number of components in "**UM**" must equal the number of components in **REFC**.
the components in "**UM**" must be a subset of the components specified in the (**REFG,REFC**) and (**G_{i,j},C_i**).
the m-set coefficient matrix in the constraint equation must be nonsingular.
4. The m-set degrees of freedom specified on this entry may not be specified on other entries that define mutually exclusive sets.
5. Rigid element identification numbers must be unique within each element type for each **MPC** set identification number.

Input Data Entry: RLOAD1

Description: Defines a frequency dependent dynamic load of the form.

$$P(f) = A [C(f) + iD(f)] e^{i(\theta - 2\pi f\tau)}$$

Format and Examples:

1	2	3	4	5	6	7	8	9	10
RLOAD1	SID	DLAGID	TC	TD					
RLOAD1	10	3	1	2					

Field	Contents
SID	Set identification number (Integer > 0)
DLAGID	Identification number of a DLAGS set which defines A, θ and τ (Integer > 0)
TC	Set identification number of TABLEDi entry which gives C(f) (Integer ≥ 0 ; TC + TD > 0)
TD	Set identification number of TABLEDi entry which gives D(f) (Integer ≥ 0 ; TC + TD > 0)

Remarks:

1. **RLOAD1** loads may be combined with **RLOAD2** loads only by specification on a **DLOAD** entry.
2. **SID** must be unique for all **RLOAD1**, **RLOAD2**, **TLOAD1** and **TLOAD2** entries.

Input Data Entry: **RLOAD2**

Description: Defines a frequency dependent dynamic load of the form.

$$P(f) = AB(f) e^{i(\phi(f) + \theta - 2\pi f\tau)}$$

Format and Examples:

1	2	3	4	5	6	7	8	9	10
RLOAD2	SID	DLAGID	TB	TP					
RLOAD2	10	6	100	101					

Field	Contents
SID	Set identification number (Integer > 0)
DLAGID	Identification of a DLAGS entry which defines A, θ and τ (Integer > 0)
TB	Set identification number of TABLED1 entry which gives B(f) (Integer > 0)
TP	Set identification number of TABLED1 entry which gives $\phi(f)$ in degrees (Integer ≥ 0)

Remarks:

1. **RLOAD2** loads may be combined with **RLOAD1** loads only by specification on a **DLOAD** entry. That is, the **SID** on a **RLOAD2** entry may not be the same as that on a **RLOAD1** entry.
2. **SID** must be unique for all **RLOAD1**, **RLOAD2**, **TLOAD1** and **TLOAD2** entries.

Input Data Entry **RROD** Rigid Rod

Description: Defines a pin-ended rod that is rigid in extension.

Format and Example:

1	2	3	4	5	6	7	8	9	10
RROD	SETID	EID	GA	GB	CMA	CMB			
RROD	1001	14	1	2	2				

Field	Contents
SETID	Multipoint constraint set identification number specified in Solution Control. (Integer > 0)
EID	Rigid Rod element identification number. (Integer > 0)
GA, GB	Grid point identification numbers of connection points. (Integer > 0)
CMA, CMB	Component number of one, and only one, dependent degree-of-freedom in the global coordinate system assigned by the element at either grid point GA or GB . (Integer 1,2 or 3, either CMA or CMB may contain the digit and the other must be blank)

Remarks:

1. The **RROD** entry is selected in the Solution Control with the **MPC=SETID** option of the **BOUNDARY** command. **THIS IS AN ENHANCEMENT TO THE NASTRAN METHOD, WHICH DOES NOT ALLOW RIGID CONNECTIONS TO BE CHANGED FOR DIFFERENT BOUNDARY CONDITIONS.**
2. The degree-of-freedom selected to be dependent must have a nonzero component along the axis of the rod; which also implies that the rod must have a finite length.
3. The m-set degrees of freedom specified on this entry may not be specified on other entries that define mutually exclusive sets.
4. Rigid element identification numbers must be unique within each element type for each **MPC** set identification number.

Input Data Entry: **SAVE**

Description: Defines a list of data base entities that are not to be purged.

Format and Examples:

1	2	3	4	5	6	7	8	9	10
SAVE	NAME1	NAME2	NAME3	NAME4	NAME5	NAME6	NAME7	NAME8	CONT
CONT	NAME9	NAME10	NAME11	-etc-					
SAVE	DVCT								

Field	Contents
NAMEi	The name of a data base entity whose contents are not to be purged.

Remarks:

1. Any number of continuations are allowed.
2. This data entry is used by the **UTPURG** utility to determine if a requested purge of an entity will take place.

Input Data Entry: SEQGP Grid and Scalar Point Resequencing

Description: Used to manually order the grid points and scalar points of the problem. The purpose of this card is to allow the user to reidentify the formation sequence of the grid and scalar points of the structural model in such a way as to optimize bandwidth.

Format and Examples:

1	2	3	4	5	6	7	8	9	10
SEQGP	ID	SEQID	ID	SEQID	ID	SEQID	ID	SEQID	CONT
CONT	ID	SEQID	-etc-						
SEQGP	5392	15.6	596	0.2	2	1.9.2.6	3		

Field	Contents
ID	Grid point identification number (Integer > 0)
SEQID	Sequenced identification number (a special number described below)

Remarks:

1. ID is any grid or scalar point identification number which is to be reidentified for sequencing purposes. The sequence number identifies a special number which may have any of the following forms where X is a decimal integer digit - XXXX.X.X.X, XXXX.X.X, XXXX.X or XXXX where any of the leading Xes may be omitted. This number must contain no embedded blanks. The leading character must not be a decimal point.
2. If the user wishes to insert a point between two already existing grid or scalar points, such as 15 and 16, for example, he would define it as, say 5392, and then use this card to insert extra point number 5392 between them by equivalencing it to, say, 15.6. All output referencing this point will refer to 5392.3. The SEQID numbers must be unique and may not be the same as a point ID which is not being changed. No extra point ID may be referenced more than once.
3. The SEQID numbers must be unique and may not be the same as a point ID which is not being changed. No extra point ID may be referenced more than once.
4. If a point ID is referenced more than once, the last reference will determine its sequence.

Input Data Entry: **SET1** Set definition for aerodynamic analysis.

Description: Defines a set of integers by a list.

Format and Examples:

1	2	3	4	5	6	7	8	9	10
SET1	SID	G1	G2	G3	G4	G5	G6	G7	CONT
CONT	G8	-etc-							

SET1	3	31	62	93	124	16	17	18	ABC
+BC	19								

Alternate Form:

1	2	3	4	5	6	7	8	9	10
SET1	SID	G1	THRU	G2					

Field	Contents
SID	Set of identification numbers (Integer > 0)
Gi	List of integers (Integer > 0)

Remarks:

1. These entries are referenced by the **SPLINE1** and **FLUTTER** data entries.
2. When using the **THRU** option, all intermediate quantities will be assumed to exist.
3. When used by **SPLINE1**, the entry refers to a list of structural grid points.
4. When used by **FLUTTER**, the entry refers to mode numbers to be omitted in the flutter analysis.

Input Data Entry: SET2 Grid Point List

Description: Defines a set of structural grid points in terms of aerodynamic macro elements.

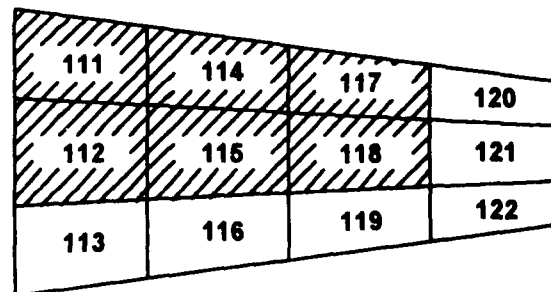
Format and Examples:

1	2	3	4	5	6	7	8	9	10
SET2	SID	SP1	SP2	CH1	CH2	ZMAX	ZMIN		
SET2	3	0.0	.73	0.0	.667	1.0	-3.51		

Field	Contents
SID	Set identification number (Integer > 0)
SP1, SP2	Lower and higher span division points defining prism containing set (1.01 > Real > -.01)
CH1, CH2	Lower and higher chord division points defining prism containing set (1.01 > Real > -.01)
ZMAX, ZMIN	Z-coordinates of top and bottom (using right-hand rule with the order of the corners as listed on a CAEROi entry) of the prism containing set (Real). Usually ZMAX > 0.0, ZMIN < 0.0

Remarks:

1. These entries are referenced by the **SPLINE1** data entries.
2. Every grid point, within the defined prism and within the height range, will be in the set. For example,



The shaded area in the figure defines the cross-section of the prism for the sample data given above. Points exactly on the boundary may be missed, hence, to get all the grid points within the area of the macro element, use **SP1** = -.01, **SP2** = 1.01, etc.

3. A zero value for **ZMAX** or **ZMIN** implies infinity is to be used.

Input Data Entry: SHAPE

Description: Defines element connectivity entries associated with a design variable.

Format and Examples:

1	2	3	4	5	6	7	8	9	10
SHAPE	SHAPEID	ETYPE	EID1	PREF1	EID2	PREF2	EID3	PREF3	CONT
CONT	EID4	PREF4	EID5	PREF5	-etc-				
SHAPE	10	CROD	12	12.0	22	1.0			

Field	Contents
SID	Shape function identification (Integer > 0)
ETYPE	Element type associated with this list (e.g., CROD)
EID1, EID2, EIDi	Element identification numbers (Integer > 0, or blank)
PREF _i	Linking factor for the associated EID (Real)

Remarks:

1. Allowable **ETYPES** are: CROD, CONROD, CSHEAR, CQDMEM1, CQUAD4, CTRIA3, CTRMEM, CBAR, CELAS1, CELAS2, CMASS1, CMASS2 and CONM2.
2. The shape function identification is referenced by the **DESVAR**s entry to connect the global variable to the shape.
3. The linking factors define a shape function to be used as the global design variable.
4. Designed properties (e.g., thicknesses) of elements listed on **SHAPE** entries will be set to unity to ensure proper shape function definition; that is, the **PREF** values define the shape to be applied to a uniform property distribution.

Input Data Entry: SPC Single-Point Constraint

Description: Defines sets of single-point constraints and enforced displacements.

Format and Examples:

1	2	3	4	5	6	7	8	9	10
SPC	SID	G	C	D	G	C	D		
SPC	2	32	436	-2.6	5		+2.9		

Field	Contents
SID	Identification number of single-point constraint set (Integer > 0)
G	Grid or scalar point identification number (Integer > 0)
C	Component number of global coordinate ($6 \geq \text{Integer} \geq 0$; up to six unique digits may be placed in the field with no embedded blanks.)
D	Value of enforced displacement for all coordinates designed by G and C (Real)

Remarks:

1. Degrees of freedom specified on this entry form members of a mutually exclusive set. They may not be specified on other entries that define mutually exclusive sets.
2. Single-point forces of constraint are recovered during stress data recovery.
3. Single-point constraint sets must be selected in Solution Control (**SPC= SID**) to be used.
4. **SPC** degrees of freedom may be redundantly specified as permanent constraints on the **GRID** entry.

Input Data Entry: **SPCADD** Single-Point Constraint Set Combination

Description: Defines a single-point constraint set as a union of single-point constraint sets defined via **SPC** or **SPC1** entries.

Format and Examples:

1	2	3	4	5	6	7	8	9	10
SPCADD	SID	S1	S2	S3	S4	S5	S6	S7	CONT
CONT	S8	S9	-etc-						

SPCADD	101	3	2	9	1				
--------	-----	---	---	---	---	--	--	--	--

Field	Contents
SID	Identification number for new single-point constraint set (Integer > 0)
Si	Identification numbers of single-point constraint sets defined via SPC or by SPC1 entries (Integer > 0; SID ≠ Si)

Remarks:

1. Single-point constraint sets must be selected in Solution Control (**SPC** = **SID**) to be used.
2. No **Si** may be the identification number of a single-point constraint set defined by another **SPCADD** entry.
3. The **Si** values must be unique.
4. **SPCADD** entries take precedence over **SPC** or **SPC1** entries. If both have the same set **ID**, only the **SPCADD** entry will be used.

Input Data Entry: SPC1 Single-Point Constraint, Alternate Form 1

Description: Defines sets of single-point constraints

Format and Examples:

1	2	3	4	5	6	7	8	9	10
SPC1	SID	C	G1	G2	G3	G4	G5	G6	CONT
CONT	G7	G8	G9	-etc-					

SPC1	3	2	1	3	10	9	6	5	ABC
+BC	2	8							

Alternate Form:

1	2	3	4	5	6	7	8	9	10
SPC1	SID	C	GID1	THRU	GID2				

SPC1	313	456	10	THRU	1000				
------	-----	-----	----	------	------	--	--	--	--

Field	Contents
SID	Identification number of single-point constraint set (Integer > 0)
C	Component number of global coordinate (any unique combination of the digits 1 through 6 (with no embedded blanks) when point identification numbers are grid points; must be null if point identification numbers are scalar points)
Gi, GIDi	Grid or scalar point identification numbers (Integer > 0)

Remarks:

1. Note that enforced displacements are not available via this entry. As many continuation entries as desired may appear.
2. Coordinates specified on this entry form members of a mutually exclusive set. They may not be specified on other entries that define mutually exclusive sets.
3. Single-point constraint sets must be selected in Solution Control (**SPC = SID**) to be used.
4. **SPC** degrees of freedom may be redundantly specified as permanent constraints on the **GRID** entry.
5. If the alternate form is used, points in the sequence **GID1** through **GID2** are required to exist.

Input Data Entry: SPLINE1 Surface Spline

Description: Defines a surface spline for interpolating out-of-plane motion for aeroelastic problems.

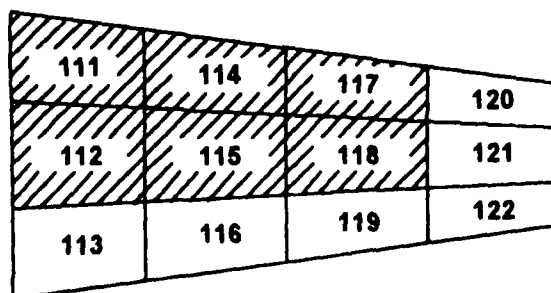
Format and Examples:

1	2	3	4	5	6	7	8	9	10
SPLINE1	EID	CP	MACROID	BOX1	BOX2	SETG	DZ		
SPLINE1	3		111	111	118	14	0.0		

Field	Contents
EID	Element identification number (Integer > 0)
CP	Coordinate system defining the spline plane (Integer ≥ 0 , or blank)
MACROID	Identification number of a CAERO _i entry which defines plane of spline (Integer > 0)
BOX1, BOX2	First and last box whose motions are interpolated using this spline (Integer > 0)
SETG	Refers to a SET _i entry which lists the structural grid points to which the spline is attached (Integer > 0)
DZ	Linear attachment flexibility (Real ≥ 0.0)

Remarks:

1. The interpolated points (k-set) will be defined by aero-cells. The sketch shows the cells for which u_k is interpolated if **BOX1** = 111 and **BOX2** = 118.



2. The attachment flexibility (units of area) is used for smoothing the interpolation. If **DZ** = 0.0, the spline will pass through all deflected grid points. If **DZ** >> (area of spline), a least squares plane fit will occur. Intermediate values will provide smoothing.
3. If no **CP** is specified, the spline plane is assumed to be the **CAERO** macro element plane.
4. The **SPLINE EID** is used only for error messages and need not be related to the macroelement identification number.

Input Data Entry: SPLINE2

Description: Defines a beam spline for interpolating panels and bodies for steady and unsteady aerodynamic analyses.

Format and Examples:

1	2	3	4	5	6	7	8	9	10
SPLINE1	EID	MACROID	BOX1	BOX2	SETG	DZ	DTOR	CID	CONT
CONT	DTHX	DTHY							
SPLINE1	1000	5000	5000	5100	10	0.	1.0	4	+SPL
+SPL	-1.								

Field	Contents
EID	Element identification number (Integer > 0)
MACROID	The identification of a CAERO1 , CAERO2 , CAERO6 or PAERO6 aerodynamic macroelement to be splined (Integer > 0)
BOX1, BOX2	The identification numbers of the first and last boxes on the macroelement to be interpolated using this spline (Integer > 0)
SETG	The identification of a SETi entry which lists the structural grid points to which the spline is attached (Integer > 0)
DZ	Linear attachment flexibility (Real ≥ 0.0)
DTOR	Torsional flexibility, $\frac{EI}{GJ}$ (Real ≥ 0.0; use 1.0 for bodies)
CID	Rectangular coordinate system which defines the y-axis of the spline (Integer ≥ 0 or blank; not used for bodies)
DTHX, DTHY	Rotational attachment flexibility. DTHX is for rotation about the x-axis; not used for bodies. DTHY is for rotation about the y-axis; used for slope of bodies. (Real)

Remarks:

1. The interpolation points (k-set) will be defined by aero-cells.
2. For panels, the spline axis is the projection of the y-axis of coordinate system **CID**, projected onto the plane of the panel. For bodies, the spline axis is parallel to the x-axis of the aerodynamic coordinate system.
3. The flexibilities are used for smoothing. Zero attachment flexibilities will imply rigid attachment, i.e., no smoothing. Negative values of **DTHX** and/or **DTHY** will imply no attachment.
4. The continuation card is optional.
5. The **SPLINE2** **EID** must be unique with respect to all other **SPLINEi** data entries, it is used only for error messages.

Input Data Entry: **SPOINT** Scalar Point List

Description: Defines scalar points of the structural model

Format and Examples:

1	2	3	4	5	6	7	8	9	10
SPOINT	ID1	ID2	ID3	ID4	ID5	ID6	ID7	ID8	

SPOINT	3	18	1	4	16	2			
--------	---	----	---	---	----	---	--	--	--

Alternate Form:

1	2	3	4	5	6	7	8	9	10
SPOINT	ID1	"THRU"	ID2						

Field

Contents

ID_i, ID1, ID2 Scalar point identification number (Integer > 0; ID1 < ID2)

Remarks:

1. If the alternate form is used, all scalar points ID1 through ID2 are defined.

Input Data Entry: SUPORT Fictitious Support

Description: Defines coordinates at which the user desires determinate reactions to be applied to a free body during analysis.

Format and Examples:

1	2	3	4	5	6	7	8	9	10
SUPPORT	SETID	ID	C	ID	C	ID	C		
SUPPORT	1000	16	215						

Field	Contents
SETID	Solution control SUPPORT set identification (Integer > 0)
ID	Grid or scalar point identification number (Integer > 0)
C	Component number (zero or blank for scalar points; any unique combination of the digits 1 through 6 for grid points)

Remarks:

1. Coordinates specified on this entry form members of a mutually exclusive set. They may not be specified on other entries that define mutually exclusive sets.
2. From one to three support coordinates may be defined on a single entry.
3. Continuation entries are not allowed.

Input Data Entry: TABDMP1 Modal Damping Table

Description: Defines modal damping as a tabular function of frequency.

Format and Examples:

1	2	3	4	5	6	7	8	9	10
TABDMP1	ID	TYPE	F ₁	G ₁	F ₂	G ₂	F ₃	G ₃	CONT
CONT	F ₄	G ₄	F ₅	G ₅	F ₆	G ₆	-etc-		
TABDMP1	3	G	0.0	0.005	1.0	0.008	2.0	0.001	ABC
+BC	2.5	.01057	2.6	.01362					

Field	Contents
ID	Table identification number (Integer > 0)
TYPE	Data word which indicates the type of damping units, G , CRIT , Q , or blank. Default is G .
F _i	Frequency value in cycles per unit time (Real ≥ 0.0).
G _i	Damping value (Real).

Remarks:

1. The **F_i** must be in either ascending or descending order but not both.
2. Jumps between two points (**F_i = F_{i+1}**) are allowed, but not at the end points.
3. At least two entries must be present.
4. Any **f_i**, **g_i** entry may be ignored by placing the BCD string **SKIP** in either of two fields used for that entry.
5. The **TABDMP1** mnemonic infers the use of the algorithm

$$g = g_t(F)$$

where **F** is input to the table and **g** is returned. The table look-up $g_T(F)$ is performed using linear interpolation within the table and linear extrapolation outside the table using the last two end points at the appropriate table end. At jump points the average $g_T(F)$ is used. There are no error returns from this table look-up procedure.

6. If **TYPE** is **G** or blank, the damping values are in structural damping units, that is, the value of **g** in (1+ig)K. If **TYPE** is **CRIT**, the damping values are in the units of fraction of critical damping, **C/C₀**. If **TYPE** is **Q**, the damping values are in the units of the amplification or quality factor, **Q**. These constants are related by the following equations:

$$C/C_0 = g/2,$$

$$Q = \left\{ \begin{array}{l} \frac{1}{\left(\frac{2C}{C_0} \right)} \\ \frac{1}{g} \end{array} \right\}$$

Input Data Entry: TABLED1

Description: Defines a tabular function for use in generating frequency-dependent and time-dependent dynamic loads.

Format and Examples:

1	2	3	4	5	6	7	8	9	10
TABLED1	ID								CONT
CONT	x ₁	y ₁	x ₂	y ₂	x ₃	y ₃	-etc-		CONT
TABDMP1	32								ABC
+BC	-3.0	6.9	2.0	5.6	3.0	5.6			

Field	Contents
ID	Table identification number (Integer > 0)

x ₁ , y ₁	Tabular entries (Real)
---------------------------------	------------------------

Remarks:

1. The **x_i** must be in either ascending or descending order but not both.
2. Jumps between two points (**x_i = x_{i+1}**) are allowed, but not at the end points.
3. At least two entries must be present.
4. Any x-y entry may be ignored by placing the string **SKIP** in either of the two fields used for that entry.
5. The generated function is:

$$Y = Y_T(X)$$

where X is input to the table and Y is returned. The table look-up $y_T(x)$ is performed using linear interpolation within the table and linear extrapolation outside the table using the last two end points at the appropriate table end. At jump points the average $y_T(x)$ is used. There are no error returns from this table look-up procedure.

Input Data Entry: **TABRND1** Power Spectral Density Table

Description: Defines Power Spectral density as a tabular function of frequency for use in Random Analysis. Referenced on the RANDPS entry.

Format and Examples:

1	2	3	4	5	6	7	8	9	10
TABRND1	ID								CONT
CONT	F1	G1	F2	G2	F3	G3	F4	G4	-etc-

TABRND1	3								ABC
+BC	2.5	.01057	2.6	.01362					

Field	Contents
ID	Table identification number (Integer > 0)
f _i	Frequency value in cycles per unit time (Real > 0.0)
g _i	Power Spectral Density (Real)

Remarks:

1. The **f_i** must be in either ascending or descending order but not both.
2. Jumps between two points (**f_i = f_{i+1}**) are allowed, but not at the end points.
3. At least two entries must be present.
4. Any **f-g** entry may be ignored by placing the BCD string **SKIP** in either of the two fields used for that entry.
5. The **TABRND1** mnemonic infers the use of the algorithm

$$g = g_T(F)$$

where F is input to the table and G is returned. The table look-up $g_T(F)$ is performed using linear interpolation within the table and linear extrapolation outside the table using the last two end points at the appropriate table end. At jump points the average $g_T(F)$ is used. There are no error returns from this table look-up procedure.

Input Data Entry: TEMP Grid Point Temperature Field

Description: Defines temperature at grid points for determination of
 (1) Thermal Loading; and (2) data recovery.

Format and Examples:

1	2	3	4	5	6	7	8	9	10
TEMP	SID	G	T	G	T	G	T		
TEMP	3	94	316.2	49	219.8				

Field	Contents
SID	Temperature set identification number (Integer > 0)
G	Grid point identification number (Integer > 0)
T	Temperature (Real)

Remarks:

1. From one to three grid point temperatures may be defined on a single entry.
2. Average element temperatures are obtained as a simple average of the connecting grid point temperatures when no element temperature data are defined.
3. For each thermal load, temperatures must be specified for all grid points using either **TEMP** or **TEMPD** entries.

Input Data Entry: **TEMPD** Grid Point Temperature Field Default

Description: Defines a temperature value for all grid points of the structural model which have not been given a temperature on a **TEMP** entry.

Format and Examples:

1	2	3	4	5	6	7	8	9	10
TEMPD	SID	T	SID	T	SID	T	SID	T	
TEMPD	1	215.3							

Field	Contents
SID	Temperature set identification number (Integer > 0)
T	Default temperature value (Real)

Remarks:

1. From one to four default temperatures may be defined on a single entry.
2. Average element temperatures are obtained as a simple average of the connecting grid point temperatures when no element temperature data are defined.
3. For each thermal load, temperatures must be specified for all grid points using either **TEMP** or **TEMPD** entries.

Input Data Entry: TF Dynamic Transfer Function

Description:

1. Used to define a transfer function of the form

$$(B_0 + B_1 p + B_2 p^2) u_d + \sum_i (A_0(i) + A_1(i) p + A_2(i) p^2) u_i = 0$$

2. May also be used as a means of direct matrix input. See Remark 3.

Format and Examples:

1	2	3	4	5	6	7	8	9	10
TF	SID	GD	CD	B0	B1	B2			CONT
CONT	G(1)	C(1)	A0(1)	A1(1)	A2(1)				

TF	1	2	3	4.0	5.0	6.0			+ABC
+ABC	13	4	5.0	6.0	7.0				

Field	Contents
SID	Set identification (Integer > 0).
GD, G(i)	Grid, scalar or extra point identification numbers (Integer > 0).
CD, C(i)	Component numbers (null or zero for scalar or extra points, any one of the digits 1 through 6 for a grid point).
B0, B1, B2, A0(i), A1(i), A2(i)	Transfer function coefficients (Real).

Remarks:

1. The matrix elements defined by this entry are added to the dynamic matrices for the problem.
2. Transfer function sets must be selected in Solution Control (**TFL = SID**) to be used.
3. The constraint relation given in Equation 1 will hold only if no structural elements or other matrix elements are connected to the dependent coordinate, u_d . In fact, the terms on the left side of Equation 1 are simply added to the terms from all other sources in the row for u_d .
4. Any number of continuations are allowed.

Input Data Entry: TIMELIST

Description: Defines a list of times at which outputs are desired.

Format and Examples:

1	2	3	4	5	6	7	8	9	10
TIMELIST	SID	TIME	TIME	TIME	TIME	TIME	TIME	TIME	CONT
CONT	TIME	TIME	-etc-						
TIMELIST	100	0.1	0.2	0.5	1.0				

Field	Contents
SID	Set identification number referenced by Solution Control (Integer > 0)
TIME	Time, (in consistent time unit) at which outputs are desired. (Real)

Remarks:

1. In order to be used, the **SID** must be referenced by Solution Control.
2. The nearest time to **TIME**, either above or below, which was used in the Transient Response analysis will be used to satisfy the output requests.
3. Any number of continuations is allowed.

Input Data Entry: **TLOAD1**

Description: Defines a time dependent function of the form:

$$P(t) = AF(t - \tau)$$

for use in a transient response problem.

Format and Examples:

1	2	3	4	5	6	7	8	9	10
TLOAD1	SID	DLAGID	TID						
TLOAD1	10	8	13						

Field	Contents
SID	Set identification number (Integer > 0)
DLAGID	Identification number of DLAGS set which defines A and τ (Integer > 0)
TID	Identification number of a TABLED1 entry which gives $F(t-\tau)$ (Integer > 0)

Remarks:

1. **SID** must be unique for all **TLOAD1**, **TLOAD2**, **RLOAD1**, and **RLOAD2** entries.

Input Data Entry: **TLOAD2**

Description: Defines a time-dependent dynamic load of the form:

$$P(t) = \begin{cases} 0 & \text{when } \bar{t} < 0 \text{ or } \bar{t} > \bar{t} - \tau \\ A \bar{t}^B e^{C\bar{t}} \cos(2\pi f\bar{t} + \theta) & \text{when } 0 \leq \bar{t} \leq \bar{t} - \tau \end{cases}$$

where $\bar{t} = t - \tau$

Format and Examples:

1	2	3	4	5	6	7	8	9	10
TLOAD2	SID	DLAGID	T1	T2	FREQ	PHASE	CTEXP	GROWTH	
TLOAD2	10	6	2.1	4.7	12.0	30.0	2.0	3.0	

Field	Contents
SID	Set identification number (Integer > 0)
DLAGID	Identification number of the DLAGS entry set which define the time invariant load A and the time delay (Integer > 0)
T1	Time constant (Real ≥ 0.0)
T2	Time constant (Real, T2 > T1)
FREQ	Frequency in cycles per unit time (Real ≥ 0.0)
PHASE	Phase angle in degrees (Real)
CTEXP	Exponential coefficient (Real)
GROWTH	Growth coefficient (Real)

Remarks:

1. **TLOAD2** loads may be combined with **TLOAD1** loads only by specification on a **DLOAD** entry.
2. **SID** must be unique for all **TLOAD1**, **TLOAD2**, **RLOAD1** and **RLOAD2** entries.

Input Data Entry TRIM Trim Variable Specification

Description: Specifies conditions for steady aeroelastic trim or nonplanar steady aerodynamic analysis.

Format and Example:

1	2	3	4	5	6	7	8	9	10
TRIM	TRIMID	MACH	QDP	TRMTYP	EFFID	VO			CONT
CONT	LABEL1	VAL1	LABEL2	VAL2	LABEL3	VAL3	LABEL4	VAL4	-etc-

TRIM	1001	0.90	1200.	LIFT	100	926.3			+ABC
+ABC	NZ	8.0	QRATE	.243	ELEV	FREE	ALPHA	FREE	

Field	Contents
TRIMID	Trim set identification number (Integer>0)
MACH	Mach number (Real ≥ 0.0)
QDP	Dynamic pressure (Real>0.0)
TRMTYP	Type of trim required (Character or blank) (See Remark 3) <ul style="list-style-type: none"> blank SUPORT controlled trim. ROLL Axisymmetric roll trim (1 DOF) LIFT Symmetric trim of lift forces (1 DOF) PITCH Symmetric trim of lift and pitching moment (2 DOF)
EFFID	Identification number of CONEFFS Bulk Data entries which modify control surface effectiveness values (Integer ≥ 0 , or blank)(Remark 2)
VO	True velocity (Real>0.0, or blank) (See Remark 12)
LABELi	Label defining aerodynamic trim parameters.
VALi	Magnitude of the specified trim parameter (Real) or the character string FREE .

Remarks:

1. The **TRIM** entry is selected in Solution Control in the **SAERO** and **NPSAERO** disciplines with the **TRIM** option.
2. All aerodynamic forces created by the control surface will be reduced to the referenced amount. For example, an **EFF1** of 0.70 indicates a 30% reduction in the forces.
3. The **TRMTYP** field has the following interpretation:

LIFT Implies that the vertical acceleration will be trimmed by one **FREE** symmetric control parameter or surface, or, the acceleration computed for some set of symmetric parameters/surfaces.

ROLL implies that the roll acceleration, **PACCEL**, will be trimmed by some one **FREE** antisymmetric control parameter or surface — OR — the acceleration computed for some set of antisymmetric parameters/surfaces. Any number of antisymmetric parameters may be fixed, but the **FREE** parameters are limited to **PACCEL** — OR — any one antisymmetric parameter or surface. For example, **PACCEL=0.0; AILERON=1.0; PRATE=FREE**

PITCH implies that the vertical acceleration, **NZ**, and the pitch acceleration, **QACCEL**, will be trimmed by no more than two **FREE** symmetric control parameters or surfaces — OR — the accelerations computed for some set of symmetric parameters/surfaces. Any number of symmetric parameters may be fixed, but the **FREE** parameters are limited to **QACCEL** and **NZ** — OR — up to two symmetric parameters or surfaces — OR — some combination. For example, **NZ=8.0g's; QACCEL=0.0; ALPHA=FREE; ELEV=FREE**

blank implies that the support DOFs are equal to the number of free parameters. Appropriate trim equations are assembled and solved.

4. Units for **QDP** are force per unit area.
5. Allowable options for **LABELi** for symmetric trim (the symmetry option is selected in Solution Control) are:

Structural Accelerations

NX Longitudinal acceleration (Remark 10)
NZ Vertical acceleration (load factor) (Remark 10)
QACCEL Pitch acceleration (Remark 11)

Aerodynamic Parameters

ALPHA Angle of attack in degrees
QRATE Pitch rate (Remark 12)
THKCAM Thickness and camber (Remark 13)
 and Control surfaces in degrees

6. Allowable options for **LABELi** for antisymmetric trim (the symmetry option is selected in Solution Control) are:

Structural Accelerations

NY Side-slip acceleration (Remark 10)
PACCEL Roll acceleration (Remark 11)
RACCEL Yaw acceleration (Remark 11)

Aerodynamic Parameters

BETA Yaw angle in degrees (Remark 14)
PRATE Roll rate (Remark 12)
RRATE Yaw rate (Remark 14)
 and Antisymmetric control surfaces in degrees

7. If **VALUEi** is a real number, the associated aerodynamic parameter or structural acceleration is set to that value. If **VALUEi** is the character string **FREE**, then the associated parameter will be determined as part of the trim analysis.

8. The number of **FREE** Values of **VALUEi** must correspond exactly to the number of unknowns in the trim analysis. If **TRMTYP** is balnk, the number of **SUPORT DOF**.
9. If **TRIMID** is referenced by an **NPSAERO** discipline, **TRMTYP** must be blank and **FREE** is not allowed for **VALUEi**.
10. For **NX**, **NY** and **NZ**, units are length per second in consistent units unless a **CONVERT/MASS** Bulk Data entry is provided. In this case, the values are dimensionless.
11. The angular accelerations, **QACCEL**, **PACCEL**, and **RACCEL**, are entered in units of radians per second per second.
12. **QRATE**, **PRATE**, and **RRATE**, are entered in units of radians per second. The velocity must be input if any of the "rate" parameters are given since its value is needed to dimensionalize the forces computed for a unit rate per velocity in the aerodynamic preface.
13. The **THKCAM** label refers to thickness and camber effects and its corresponding value is usually set to 1.0. Non-unit values of the **THKCAM** parameter are available only to provide added generality.
14. Any control surfaces, trim parameters, or structural accelerations not specified on the **TRIM** entry will not participate in the analysis: they will be given fixed values of 0.0. This includes **THKCAM**.
15. Refer to the **STATIC AEROELASTIC TRIM** Application Note for more information.

Input Data Entry: TSTEP

Description: Defines time step intervals at which a solution will be generated and output in transient analysis.

Format and Examples:

1	2	3	4	5	6	7	8	9	10
TSTEP	SID	N (1)	DT (1)	NO (1)					CONT
CONT		N (2)	DT (2)	NO (2)					CONT
TSTEP	2	10	.001	5					+ABC
+ABC		9	0.01	1					

Field	Contents
SID	Set identification number (Integer > 0)
N (i)	Number of time steps of value DT (i) (Integer ≥ 2)
DI (i)	Time increment (Real > 0.0)
NO (i)	Skip factor for output (every NO (i) th step will be saved for output) (Integer > 0)

Remarks:

1. **TSTEP** entries must be selected in the Solution Control (**TSTEP=SID**).
2. Note that the entry permits changes in the size of the time step during the course of the solution. Thus, in the example shown, there are 10 time steps of value .001 followed by 9 time steps of value .01. Also, the user has requested that output be recorded for $t = 0.0, .005, .01, .02, .03$, etc.

Input Data Entry: **VSDAMP**

Description: Specifies values of g and/or ω_3 to generate either viscous damping that has the same damping forces as structural damping of magnitude g at the frequency ω_3 or to specify the structural damping g (see Remarks 3 and 4)

Format and Examples:

1	2	3	4	5	6	7	8	9	10
VSDAMP	SID	G	ω_3	SID	G	ω_3			
VSDAMP	100	0.005	15.0						

Field	Contents
SID	Set identification number (Integer > 0)
G	Damping value (Real)
ω_3	Frequency value in Hertz (Real ≥ 0.0)

Remarks:

1. The setid is selected by the **DAMPING=n** command in Solution Control.
2. Up to two values of g and ω_3 can be defined on a single entry.
3. If ω_3 is zero, g will be used to generate a complex stiffness matrix \mathbf{K} of the form $\mathbf{K} = (1 + ig) \mathbf{K}$
4. If ω_3 is nonzero, a viscous damping matrix of the form $\mathbf{B} = \left[\frac{g}{2\pi\omega_3} \right] \mathbf{K}$ is generated.

5. ASTROS OUTPUT FEATURES

In a software system the magnitude of ASTROS, the amount of data that may be of interest to the user is very large. In multidisciplinary optimization, the quantity of data is even larger and the expense involved in its computation even more critical. It is worthwhile, therefore, to limit the amount of output to a minimum and to provide a mechanism for the user to select those data that are of importance in each particular case. Chapter 3 of this report documented one mechanism provided in ASTROS to select particular iterations, disciplines, subcases and response quantities: that of the Solution Control output request. This section endeavors to present the totality of output options available to the ASTROS user. The system controlled outputs from the engineering modules are described in order to establish a familiarity with an ASTROS output listing. This is followed by a more complete description of output from each Solution Control request than is contained in Chapter 3, with different disciplines, elements, design constraints and node types accounted for in some detail. These represent the outputs that are fully supported by the ASTROS software and require little or no user intervention to obtain. The presentation of these features assumes that the standard executive sequence is used. If the user substantially modifies the standard sequence (to the point where certain modules are not called), some or all of the presented output features may no longer be available.

The more advanced forms of user output requests are also presented in this section. The most basic of these forms involve changing the engineering module print control levels through the use of the DEBUG packet. Then, the MAPOL addressable print utilities are presented. The use of these utilities, in conjunction with the general versatility of the MAPOL language, provides the user with the capacity both to look at existing data and to compute and view additional data. In fact, these options enable the user to obtain virtually any data that reside on the data base or that can be computed and stored on the data base. Finally, a quick overview of the Interactive CADDB Environment (ICE) is given. The ICE program provides for complete Standard Query Language interactive queries on the CADDB entities.

5.1. SYSTEM CONTROLLED OUTPUT

Many of the engineering and executive system program units write data to the ASTROS output listing automatically. As enumerated in the introduction to this section, output of this nature in ASTROS is very limited, but sufficient amounts exist to justify a brief presentation of the data and their formats. It is also useful to present the basic ASTROS listing in order to facilitate contrasting it to listings containing user selected output quantities. The first page of ASTROS output is the title page showing the version number, date and host machine of the user's system. Each page of output following the solution control listing will be labeled with six lines of header information including the user selected title, subtitle and label. The version number, date and, if applicable, the design iteration number will also appear in the header of each page.

5.1.1. Default Output Printed by Modules

The **DEBUG** packet echo and the **ASSIGN DATABASE** entries, shown in Table 18, are the first output following the title page. Immediately following these, the solution control commands will be echoed to the output listing. This listing is helpful in identifying the particular disciplines and cases selected in the run. The multidisciplinary nature of **ASTROS** requires further output labeling; therefore, in addition to the solution control summary, the **BOUND** module writes a summary of selected disciplines for each boundary condition at the top of the boundary condition loop, presented in Table 19. It indicates all disciplines and most discipline options in the current boundary condition to assist the user in determining the particular path that will be taken through the standard **MAPOL** sequence. A similar print, Table 20, from the **ABOUND** module appears at the top of the sensitivity phase boundary condition loop to indicate the nature of the active boundary conditions and active design constraints.

The next set of output, Table 21, comes from the bandwidth minimizer. It details the method selected, numbers of grids and elements in the model and the values of the measures of merit in the resequencing of grid points.

Active constraint information is provided in the Active Constraint Summary from the **ACTCON** module. It indicates the total number of constraints considered active according to the current constraint deletion criteria. The user may select a complete listing of the active constraints through the **PRINT DCONSTRAINT** solution control option, but may not suppress the table header indicating the number of constraints retained of the total number applied. This number is computed even if the current design is considered to be the converged optimum. A summary of the convergence criteria and of the critical constraint value is included in the Active Constraint Summary header, illustrated in Table 22, if the approximate problem was considered converged following the preceding redesign step.

Each redesign step is summarized in a small table, shown in Table 23, entitled the Approximate Optimization Summary. It indicates the optimization method used in resizing and the changes in three measures of convergence. The first measure is the change in the value of the objective function during the solution of the approximate optimization problem. The second is the change in the Euclidean norm of the design variable vector and finally, the maximum absolute change in any component of the design variable vector. Each of the values are computed as an absolute change and a percentage change. These values are then printed. You may compare the first two percentage values against your input convergence limit, denoted **UPPER BOUND PERCENT MOVE**, to determine which (if either) is greater than the limit. If either value is greater, the approximate problem will not be considered converged, otherwise it will be. A message indicating the state of convergence closes the Approximate Optimization Summary.

The last default design print, Table 24, is generated by the **ACTCON** module on the final design iteration. The **ACTCON** module prints out the design iteration history. The iteration history includes statistics summarizing each *approximate* optimization problem and shows the increments in the objective function. All values in this table are associated with the approximate problem. Since weight in **ASTROS** is explicitly linear in the design variables, the objective function values are exact.

The final default outputs are a trailer indicating the status of the termination (either with or without errors), the date and the time the run was completed and an execution timing summary. The timing summary, shown in Table 25, indicates the CPU time spent in each phase of the execution. The

Table 18. DEBUG and ASSIGN DATABASE Output

```

*****
::::::::::::::::::::::::::::::::::::::::::::
::
::      AUTOMATED STRUCTURAL OPTIMIZATION SYSTEM
::
::
::      *****
::      *       *       *       *       *       *
::      *   .   *   .   .   .   .   .   .
::      *   .   *   .   .   .   .   .   .
::      *   .   *   .   .   .   .   .   .
::      *   .   *   .   .   .   .   .   .
::      *   .   *   .   .   .   .   .   .
::
::
::              VERSION 9.0
::
::
::          IBM RISC SYSTEM/6000
::          JUL 01, 1992
::
::
::::::::::::::::::::::::::::::::::::::::::::

***** ASTROS DEBUG PACKET ECHO *****

DEBUG KEY "LOGBEGIN " HAS BEEN SELECTED
DEBUG KEY "LOGMODULE " HAS BEEN SELECTED
DEBUG KEY "MATRIX " HAS BEEN SELECTED

***** ASTROS ASSIGN DATABASE COMMAND ECHO *****

*...10...**...20...**...30...**...40...**...50...**...60...**...70...**...80...*

ASSIGN DATABASE COMB SHAZAM NEW

*...10...**...20...**...30...**...40...**...50...**...60...**...70...**...80...*

DATA BASE NAME           = COMB
DATA BASE PASSWORD = SHAZAM
DATA BASE STATUS        = NEW
USER PARAMETERS ARE:
** NONE GIVEN **

```

Table 19. Boundary Condition Summary

BOUNDARY CONDITION SUMMARY FOR BOUNDARY CONDITION 2

MATRIX DEFINITION SUMMARY:

THE PHYSICAL SET CONTAINS	1948 DEGREES OF FREEDOM (DOFS)
	1948 PHYSICAL DOFS ARE STRUCTURAL
AND	0 PHYSICAL DOFS ARE EXTRA POINTS

THERE ARE \dots INDEPENDENT MULTIPoint CONSTRAINT DOFS LEAVING
 \dots INDEPENDENT DOFS

THERE ARE 1560 SINGLE POINT CONSTRAINT DOFS LEAVING
2373 FREE DOFS

THE FREE DOFS ARE REMOVED USING STATIC CONDENSATION
THERE ARE 2.1 OMITTED DOFS LEAVING
15. ANALYSIS SET DOFS
OF WHICH ARE "SUPPORTED" LEAVING
15. DOFS LEFT OVER

```
DISCIPLINE SUBPASE SUMMARY:
*** STATISTICS HAVE BEEN SELECTED ***
    5 SUBPASE(S) ARE DEFINED BY SOLUTION CONTROL.
*** FINITER HAS BEEN SELECTED ***
    A REAL EIGENANALYSIS WILL ALSO BE DONE IF NOT ALREADY SELECTED
    1 SUBPASE(S) ARE DEFINED BY SOLUTION CONTROL.
```

Table 20. Active Boundary and Constraint Summary

SENSITIVITY SUMMARY FOR BOUNDARY CONDITION 1

4 FLUTTER CONSTRAINTS

4 TOTAL ACTIVE CONSTRAINTS FOR THIS BOUNDARY CONDITION.

SENSITIVITY SUMMARY FOR BOUNDARY CONDITION 2

2 TSAI-WU STRESS CONSTRAINTS ON STATIC SUBCASE 1

6 TSAI-WU STRESS CONSTRAINTS ON STATIC SUBCASE 2

4 FLUTTER CONSTRAINTS

12 TOTAL ACTIVE CONSTRAINTS FOR THIS BOUNDARY CONDITION.

Table 21. Resequencing Summary

*** SUMMARY OF AUTOMATIC RESEQUENCING ***

METHOD SELECTED CRITERION	CM RMS WAVEFRONT	
	BEFORE	AFTER
BANDWIDTH	648	60
PROFILE	25960	13960
MAXIMUM WAVEFRONT	40	60
AVERAGE WAVEFRONT	39.453	36.413
RMS WAVEFRONT	40.632	38.159
NUMBER OF GRID POINTS		658
MAXIMUM NODAL DEGREE		14
NUMBER OF MPC EQUATIONS PROCESSED		12
ELEMENTS PROCESSED		
	CSHEAR	565
	CTRMEM	40
	CONROD	305
	CONM2	35
	CBAR	152
	CQUAD4	662
TOTAL ELEMENTS		1759

Table 22. Active Constraint Summary

```

SUMMARY OF ACTIVE CONSTRAINTS
AFTER ANALYSIS 4 OF A MAXIMUM 16
12 CONSTRAINTS RETAINED OF 60 APPLIED

THE APPROXIMATE OPTIMIZATION PROBLEM WAS CONVERGED WITH
FEASIBLE CONSTRAINT CRITERIA (CTLMIN).... 5.00000E-04 AND
ACTIVE CONSTRAINT CRITERIA (CTL)..... 7.50000E-04

CURRENT MAXIMUM CONSTRAINT VALUE.... -4.97155E-04

DO TERMINATE ..... -2.25000E-03 < -4.97155E-04 <= 1.00000E-03

*** ASTROS OPTIMIZATION HAS CONVERGED ***

.....
* CONSTRAINT RETENTION ALGORITHM SUMMARY *
* RFAC = 3.000, EPS = -.100, NDV = 4 *
*
* # OF CONSTRAINTS RETAINED BY RFAC = 12 *
* CUTOFF CONSTRAINT VALUE = -.981 *
*
* # ADDED WITH VALUES GREATER THAN EPS = 0 *
*
* # OF ADDITIONAL MINIMUM THICKNESS *
* CONSTRAINTS RETAINED ONLY FOR *
* CONTROLLING MOVE LIMITS (DCONTHK) = 0 *
*
.....

COUNT CONSTRAINT VALUE CONSTRAINT TYPE TYPE COUNT BOUNDARY ID SUBCASE ELEMENT TYPE EID/LAYR
1 7.09291E-02 UPPER BNI LEFT EFFECT 1 1 1 N/A N/A
2 4.97155E-04 DISPLACEMENT 1 1 1 N/A N/A
3 4.00266E-01 VON MISES STRESS 1 1 1 QDMEM1 13
4 8.11897E-01 VON MISES STRESS 2 1 1 QDMEM1 14
5 8.41767E-01 VON MISES STRESS 3 1 1 QDMEM1 16
6 5.62036E-01 VON MISES STRESS 4 1 1 QDMEM1 17
7 4.38954E-01 VON MISES STRESS 5 1 1 QDMEM1 20
8 4.14866E-01 VON MISES STRESS 6 1 1 QDMEM1 21
9 8.52444E-01 VON MISES STRESS 7 1 1 QDMEM1 23
10 5.76223E-01 VON MISES STRESS 8 1 1 QDMEM1 26
11 9.77224E-01 VON MISES STRESS 10 1 1 ROD 2
12 9.81033E-01 VON MISES STRESS 20 1 1 SHEAR 29

```

Table 23. Approximate Optimization Summary

```

**** ASTROS APPROXIMATE OPTIMIZATION SUMMARY ****
*** ITERATION 1 ***
** RESIZING METHOD = MATHEMATICAL PROGRAMMING **
** DESIGN VAR. MOVE LIMIT = 2.000000 *
* UPPER BOUND PERCENT MOVE = 1.000000 PERCENT *
* CRITERION 1: OBJECTIVE CHANGE *
* CURRENT VALUE = 4.3531E+01 *
* PREVIOUS VALUE = 2.7840E+01 *
* DELTA = 1.5691E+01 *
* PERCENT MOVE = 56.3603 *
* CRITERION 2: DESIGN VECTOR MOVE *
* NORM OF X - X0 = 1.3076E+00 *
* EUCLIDEAN NORM OF X0 2.3200E+00 *
* PERCENT MOVE 56.3603 *
* CRITERION 3: DESIGN VARIABLE MOVE *
* MAXIMUM MOVE 3.6350E-01 *
* AT DESIGN VARIABLE 3 *
* CURRENT VALUE = 1.2339E+00 *
* PREVIOUS VALUE = 8.7000E-01 *
* PERCENT MOVE 41.8275 *
* THE APPROXIMATE PROBLEM IS NOT CONVERGED *
.....

```

Table 24. Design Iteration History

ASTROS DESIGN ITERATION HISTORY									
ITERATION NUMBER	OBJECTIVE FUNCTION VALUE	NUMBER FUNCTION EVAL	NUMBER GRADIENT EVAL	NUMBER RETAINED CONSTRAINTS	NUMBER ACTIVE CONSTRAINTS	NUMBER VIOLATED CONSTRAINTS	NUMBER LOWER BOUNDS	NUMBER UPPER BOUNDS	APPROXIMATE PROBLEM CONVERGENCE
1	1.25894E+04	(INITIAL FUNCTION VALUE)							
2	7.12705E+03	31	4	18	1	0	0	0	NOT CONVERGED
3	6.36273E+03	30	6	18	1	0	1	1	NOT CONVERGED
4	6.08681E+03	39	8	18	1	0	3	1	NOT CONVERGED
5	5.89348E+03	47	11	18	1	0	4	0	NOT CONVERGED
6	5.74943E+03	56	13	18	1	0	4	0	NOT CONVERGED
7	5.62364E+03	38	9	18	1	0	4	1	NOT CONVERGED
8	5.50224E+03	40	10	18	1	0	4	1	NOT CONVERGED
9	5.38496E+03	24	7	18	1	0	4	1	NOT CONVERGED
10	5.27604E+03	36	8	18	2	0	4	1	NOT CONVERGED
11	5.18694E+03	43	11	18	2	0	4	1	NOT CONVERGED
12	5.14224E+03	33	5	18	2	0	4	1	NOT CONVERGED
13	5.13861E+03	10	2	18	2	0	4	1	NOT CONVERGED
14	5.13618E+03	10	1	18	2	0	4	1	NOT CONVERGED
15	5.11049E+03	18	3	18	2	0	4	1	CONVERGED

THE FINAL OBJECTIVE FUNCTION VALUE IS:

FIXED =	0.00000E+00
+ DESIGNED =	5.11049E+03

TOTAL =	5.11049E+03

elapsed clock time is shown upon leaving each phase of the MAPOL execution. This summary is useful in determining where a problem may have occurred and in confirming that the proper path was taken through the MAPOL sequence. It is, of course, also useful as an indication of the relative CPU costs of each phase of execution.

5.1.2. Error Message Output

Error messages can be printed from virtually all the modules of the ASTROS system as well as from the data base management software. Data base errors should not occur unless the user has modified or otherwise written a special MAPOL sequence, incorrectly assigned file names or used other incorrect or inconsistent data base information. Typically, data base errors cause immediate termination of the execution. The system administrator should be able to assist in solving these errors which, it is felt, will most likely be caused by incorrect use of the system or by incorrect system installation. The ASTROS Programmer's Manual contains further information on the causes of particular data base errors.

The standard ASTROS error messages are printed by the **UTMVRT** utility module and represent error checks that the modules are coded to perform or errors that may cause problems in the current module's algorithm. As much as possible, these error messages are intended to be standalone in that the user should be able to interpret the message without referring to the Programmer's Manual. There are four different levels of errors that can occur in ASTROS, each labeled differently when printed:

(1) System Fatal Message

These messages come about due to errors or inconsistencies in the system definitions. Usually, these relate to erroneous input to the system generation utility, SYSGEN, or are a result of using an outdated system data base. You should contact your system adminis-

Table 25. ASTROS Execution Summary

*** ASTROS TERMINATED ***		
*** 02/16/94 11:45:50 ***		

ASTROS TIMING SUMMARY		
ELAPSED TIME	TOTAL CPU	STEP CPU
00:00:00	00:00:00.0	*** BEGIN ASTROS ***
00:00:01	00:00:01.9	BEGIN FREEFACE MODULES
00:00:14	00:00:16.4	ELEMENT MATRIX GENERATION
00:04:43	00:05:07.4	NON PLANAR STEADY AERODYNAMICS
00:04:43	00:05:07.4	PHASE 1 ELEM. MATRIX ASSEMBLY
00:05:04	00:05:37.8	PHASE 1 STATIC LOADS GENER.
00:05:05	00:05:38.0	STEADY AERODYNAMICS
00:05:05	00:05:38.1	UNSTEADY AERODYNAMICS
00:07:58	00:09:15.1	*****
00:07:58	00:09:15.1	BEGIN OPTIMIZATION
00:07:58	00:09:15.1	*****
00:07:58	00:09:15.1	DESIGN ITERATION 1
00:08:27	00:09:24.3	BOUNDARY CONDITION 1
00:08:29	00:09:26.1	MPC REDUCTION
00:09:03	00:09:54.3	SFC REDUCTION
00:09:11	00:09:58.5	STATIC CONDENSATION
00:10:12	00:06:52.9	>>>DISCIPLINE: NORMAL MODES
00:10:12	00:06:54.8	>>>DISCIPLINE: FLUTTER
00:10:13	00:07:04.0	DATA RECOVERY
00:10:33	00:07:04.1	STATIC CONDENSATION RECOVERY
00:10:34	00:07:04.8	SFC RECOVERY
00:10:34	00:07:05.0	MPC RECOVERY
00:10:34	00:07:05.4	CONSTRAINT EVALUATION
00:10:34	00:07:05.4	OUTPUT PROCESSING
00:10:35	00:07:05.5	BOUNDARY CONDITION 2

00:12:34	00:08:48.0	CONSTRAINT EVALUATION
00:13:18	00:09:07.8	OUTPUT PROCESSING
00:14:00	00:09:07.9	SENSITIVITY ANALYSIS
00:14:47	00:09:17.7	DESIGN MODULE
00:14:52	00:09:16.1	*****
00:14:52	00:09:16.1	DESIGN ITERATION 2

00:10:15	00:04:51.3	*** END ASTROS ***

trator to effect a correction. Hopefully, these errors will rarely occur and should never occur in an unmodified ASTROS system.

(2) User Information Message

These messages are written when the system encounters data that may represent an input error or may later generate a problem but that may only be a special user input that falls outside the expected range. Usually, these messages can be ignored. This is the least serious type of user message in ASTROS.

(3) User Warning Message

These messages are written when the system encounters data that are incorrect but which may not cause termination. In some cases, this level of error is issued to signify that the system will continue to search for errors but will terminate abnormally following the search.

(4) **User Fatal Message**

These messages are written when the system encounters data that are in error to the extent that continuation is impossible. The system will terminate execution either immediately or after some minor clean up.

If the user is unable to decipher the error message, the following steps can be helpful in determining the source of the error:

- (1) Check the timing summary with the LOGBEGIN and LOGMODULE options in the DEBUG packet against the MAPOL sequence path to determine which module generated the error message. Also, check the SYSGEN output to determine the module that wrote the message. Note that the "message number" is included in the error message print if the message is a standard one and the message number can be used to trace the module that uses the message.
- (2) Check the Programmer's Manual documentation for the relevant module to determine the error checks it performs and to get further information on the source of the error.

5.2. SOLUTION CONTROL OUTPUT OPTIONS

This subsection presents a detailed description of the output quantities that can be selected through the solution control packet. These quantities fall into five categories: (1) element; (2) nodal; (3) design; (4) eigenvalues for flutter and normal modes; and (5) aeroelastic trim quantities. Each of these categories is presented in the separate subsections that follow.

The **PRINT** and **PUNCH** solution control commands are used to request the desired output quantities. These commands have three groups of options: subset options, quantity options and form options. These options are fully described in Chapter 3 of this report, but one point must be stressed: subset options play an extremely important role in ASTROS output requests. Subset options allow the user to identify the set of iterations or subcases to which the print selection will apply. This selection is necessary because many disciplines (**MODES**, for example) generate more than one subcase (eigenvector) with a single solution control directive. The critical point is that the default selection for subcases is that there be no output. In other words, if there are no subcases selected, ASTROS will, by default, *print nothing*.

Unlike NASTRAN, ASTROS has no options to reorder the output. The multidisciplinary nature of ASTROS completely negates the utility of NASTRAN's **SORT1** and **SORT2** options, and any other sort options become impossibly complex very quickly. Instead, a reasonable, fixed sort was established in which each boundary condition is treated separately and in the order given in the solution control packet. If the standard sequence is used, the response quantities will appear in the following order within each optimize or analyze boundary condition:

- (1) Non-planar steady aerodynamics output.
- (2) Steady aerodynamic trim parameters.
- (3) Flutter roots and flutter mode shape modal participation factors -- note that the mode shape is only available if flutter has occurred and if the **FLUTTER** discipline is within an **ANALYZE** boundary condition.
- (4) Applied **LOAD** print requests.
- (5) The "displacement" nodal response quantities: **DISPLACEMENTs**, **VELOCITYs**, and **ACCELERATIONS**.

- (6) Element response quantities in the order **STRESS**, **STRAIN**, **FORCE** and **STRAIN ENERG** for each subcase, elements are processed alphabetically within each quantity type.

In the **OPTIMIZE** subpacket, these data are followed by the selected design and resizing prints in following order:

- (7) Active constraint summary (either the default abbreviated print or the full print if the **DCONSTRAINT** print option is selected).
- (8) The print of the global and then local design variables representing the current design peding on the **GDESIGN** and **LDESIGN PRINT** requests. On the final design iteration, th iteration history precedes the design variable output by default.

Within each response quantity's print module, the disciplines are not treated in the order given in solution control packet; instead, they are treated, where applicable, in the following order:

- (A) **NPSAERO**
- (B) **STATICS**
- (C) **MODES**
- (D) **SAERO**
- (E) **TRANSIENT**
- (F) **FREQUENCY**
- (G) **BLAST**

The subcases within each discipline are treated in the order given in the solution control packet. In case of **MODES**, the eigenvectors are ordered in increasing eigenvalue order. **TRANSIENT**, **FREQUENCY**, **BLAST** subcases are ordered in increasing time or frequency step.

5.2.1. Element Response Quantities

ASTROS has two basic forms of elements: aerodynamic elements and structural elements. aerodynamic element is defined as a "box" of an aerodynamic macroelement, e.g., wing component fuselage segment. The nature of the macroelement varies among both aerodynamic models and aerodynamic components within each model. In general, however, a box is the smallest subdivision o aerodynamic component for which data (e.g., pressures, forces, and moments) are computed. Structural elements are either metric elements, which connect structural node points (grids); scalar elements, w connect pairs of degrees of freedom or pairs of scalar points; or mass elements. Table 26 shows the li aerodynamic and structural elements in ASTROS for which element output exist. The following sul tions document the quantities that are available as output for each of these elements. The struc mass elements are not included in this table since they have no element response quantities. NASTRAN User's Manual (Reference 2) was used as a major resource in writing this section and the is referred to it for additional information on the structural elements.

Structural element output is available for all disciplines that result in a real displacement. This includes **STATICS**, **MODES**, **TRANSIENT**, **BLAST**, and **SAERO** analyses. Complex displacement (from **FLUTTER** and **FREQUENCY** analyses) result in computation of the selected (complex) element response quantities, but their formatted print is not available except through executive sequence utilities described in Subsection 3.4. For all disciplines in ASTROS, the solution control print op **STRESS**, **STRAIN**, **FORCE**, and **ENERGY** are used to select print of the structural element quantities. **AIRDISP** and **TPRESSURE** options are used for aerodynamic element quantities. Each of these

Table 26. Aerodynamic and Structural Elements in ASTROS

AERODYNAMIC	STRUCTURAL
CAERO1 CAERO2 CAERO6 PAERO6	CBAR CELAS1, CELAS2 CIHEX1, CIHEX2, CIHEX3 CROD, CONROD CQDMEM1, CTRMEM CTRIA3, CQUAD4

options selects either **ALL**, **NONE** or an integer set identification number that refers to one or more **ELEMLIST** bulk data entries specifying which elements are to have output computed and printed. Chapter 3 contains the complete description of the solution control print command. Each output is carefully labeled as to its boundary condition number, discipline generating the response field and load condition, mode number, time step, frequency step or flight condition represented by the output.

5.2.1.1. Aerodynamic Element Output

The solution control **PRINT** option **TPRESSURE** provides the trimmed pressures on the aerodynamic boxes for **SAERO** and **NPSAERO**. The **NPSAERO** trimmed pressures are printed to the output file. For **SAERO**, the trimmed pressures are computed and stored in the relational entity **OAGRDLOD**. The **AIRDISP** print option is available for the **SAERO** discipline and provides the out-of-plane displacements and stream-wise slopes of the aerodynamic boxes that coorespond to the structural displacements. These data are computed and stored on the relational entity **OAGRDDSP**.

Aerodynamic geometry data are computed and stored by default to a set of relational entities that parallel the structural model. These data forms are designed primarily for model checkout of the **SAERO** and **NPSAERO** models using existing FE preprocessors that support NASTRAN style input data. These relations are: **AEROGEOM** which supplies the GRID-like data and **CAROGEOM** which provides connectivity data for the boxes in a ROD or QUAD4 form. The ROD is used to model the outline of the airfoils and QUAD4's are used to model the boxes.

For unsteady aerodynamics, the box-on-box aerodynamic forces are only available through the **DEBUG/UNSTEADY** and **DEBUG/AMP** options (see Chapter 1). The geometry data are not available.

5.2.1.2. Bar Element Output

The BAR element includes extension, torsion, bending in two perpendicular planes and the associated shears. The shear center is assumed to coincide with the elastic axis. The BAR element coordinate system is shown in Figure 5. The orientation of the BAR element is described in terms of two reference planes defined through the use of the orientation vector(*v*) as shown in that figure. The positive directions for the element forces are shown in Figure 6. Additional information on the structural element is contained in Section 5 of the ASTROS Theoretical Manual.

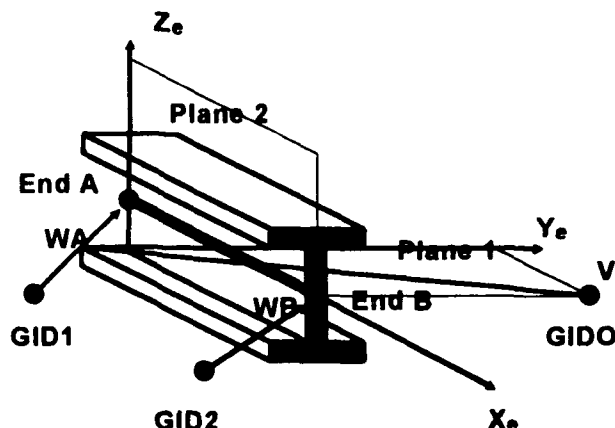


Figure 5. BAR Element Coordinate System

Stresses, strains, forces and strain energies are available as output for the BAR element through the **STRESS**, **STRAIN**, **FORCE**, and **ENERGY** solution control print options. The following element forces are output on request:

- (1) Bending moments at each end in both reference planes.
- (2) Shear forces in each reference plane.
- (3) Average axial force.
- (4) Torque about the bar axis.

The following element stresses and strains in the element coordinate system are output on request:

- (1) Average axial stress or strain.
- (2) Extensional stress or strain due to bending at 4 points on the cross-section at each end.
- (3) Maximum and minimum stress or strain at each end.
- (4) Stress margins of safety for the element in both tension and compression.

Tensile stresses and strains are given a positive value while compressive stresses and strains are given a negative value. The bending contribution to the stresses are always computed at the four points on the element cross-section that were specified on the connectivity entry for the BAR element. This means that the safety margins are computed using all eight stress values even if all 4 stress points at each end are the same and/or coincide with the element axis. Also, margins of safety are printed even if no stress limits were given on the material entry. In these cases, a very large value for the margin of safety is used to indicate that no limits were specified. In addition, ASTROS fully supports strain output for the BAR element. Strain energies may also be requested for the BAR element. The strain energy printout (which is identical for all ASTROS structural elements) is patterned after that in NASTRAN. It shows the total strain energy for the given displacement field, the strain energy in each selected element and the total strain energy for all the elements of a given type, e.g., all the BAR elements. Examples of each of these outputs are shown in Table 27.

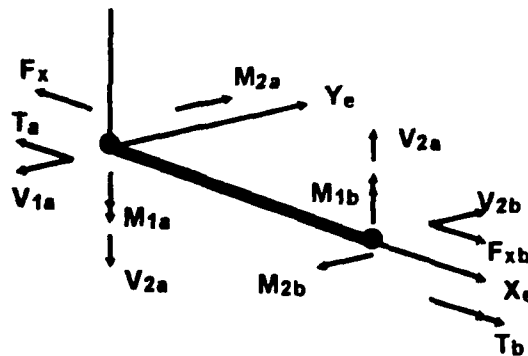


Figure 6. BAR Element Forces Sign Conventions

Table 27. BAR Element Output Quantities

ASTROS VERSION 9.0 03/03/93 ELEM 106 FINAL ANALYSIS SEGMENT									
TRANSIENT ANALYSIS: BOUNDARY 1, TIME = 4.999999E+00									
(BAR)									
ELEMENT ID.	SA1 SB1	SA2 SB2	SA3 SB3	SA4 SB4	AXIAL STRESS	SA-MAX SB-MAX	SA-MIN SB-MIN	MISC.1 MISC.2	
106	-2.810616E+03	0.000000E+00	-2.810616E+03	2.810616E+03	0.000000E+00	2.810616E+03	-2.810616E+03	1.7E+08	
	7.498176E-02	0.000000E+00	7.498176E-02	-7.498176E-02		7.498176E-02	-7.498176E-02	1.7E+08	
ASTROS VERSION 9.0 03/03/93 ELEM 106 FINAL ANALYSIS SEGMENT									
TRANSIENT ANALYSIS: BOUNDARY 1, TIME = 4.999999E+00									
(BAR)									
ELEMENT ID.	SA1 SB1	SA2 SB2	SA3 SB3	SA4 SB4	AXIAL STRAIN	SA-MAX SB-MAX	SA-MIN SB-MIN		
106	-1.963023E-04	0.000000E+00	-1.963023E-04	1.963023E-04	0.000000E+00	1.963023E-04	-1.963023E-04		
	5.857950E-10	0.000000E+00	5.857950E-10	-5.857950E-10		5.857950E-10	-5.857950E-10		
ASTROS VERSION 9.0 03/03/93 ELEM 106 FINAL ANALYSIS SEGMENT									
TRANSIENT ANALYSIS: BOUNDARY 1, TIME = 4.999999E+00									
(BAR)									
ELEMENT ID.	BEND-MOMENT PLANE 1	BEND-MOMENT PLANE 2	BEND-MOMENT PLANE 1	BEND-MOMENT PLANE 2	AXIAL SHEAR	PLANE 1 PLANE 2	PLANE 1 PLANE 2	PLANE 1 PLANE 2	PLANE 1 PLANE 2
106	0.000000E+00	3.195801E+00	0.000000E+00	-9.536743E-06	0.000000E+00	6.391611E-01	0.000000E+00	0.000000E+00	0.000000E+00
ASTROS VERSION 9.0 03/03/93 ELEM 106 FINAL ANALYSIS SEGMENT									
TRANSIENT ANALYSIS: BOUNDARY 1, TIME = 4.999999E+00									
ELEMEN T STRAIN ENERGIES									
BAR	ELEMENTS			TOTAL ENERGY OF ALL ELEMENTS IN THE SUB-AS			4.168508E+01		
	ELEMENT ID			STRAIN ENERGY			PERCENT OF TOTAL		
	101			1.430531E-01			0.259322		
	102			4.057112E-04			.924499		
	103			4.288774E-04			9.773748		
	104			1.405888E-01			329.95667		
	105			1.847305E-05			4.4204154		
	106			5.228124E-04			11.913216		
BAR	ELEMENTS			SURTOTAL			100.000000		

5.2.1.3. ELAS Element Output

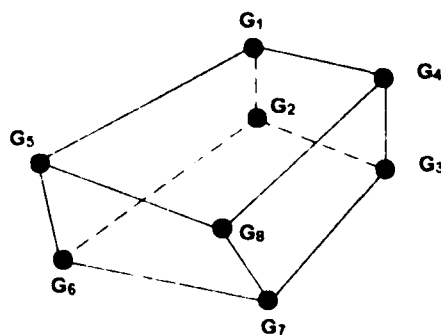
The ELAS element is a scalar spring element which relates the displacements at a pair of scale points or degrees of freedom or that relates a single degree of freedom to a ground state. The element force and strain energy are directly available for the element and the user can, if desired, input a scalar quantity that relates the "stress" in the element to the displacement(s) of the connected degree(s) of freedom. On output, these values will be printed for each output request for each selected ELAS element. Strains have no meaning for the scalar spring element and any such requests will be ignored without warning. Element strain energies, however, are available for the element and are computed from the spring constant and the nodal displacement(s). The strain energy print for the ELAS is identical to that for the BAR element and includes a breakdown by element and by element type. If no scalar value is given for the element stress but the stress value is requested, a value of zero will be computed and printed for the response quantity with no warnings given.

5.2.1.4. IHEX1 Element Output

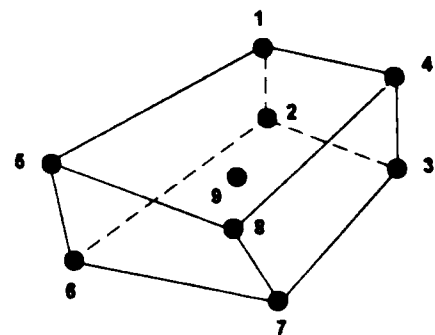
The IHEX1 element is a linear isoparametric solid hexahedron element with three extensional degrees of freedom for each of its eight nodes.

Stresses, strains, and strain energies are available as output for the IHEX1 element through the **STRESS**, **STRAIN**, and **ENERGY** solution control print command options. Force output is not available for the IHEX1 element. On request, the following stresses and strains are output in the basic coordinate system at the center and at each corner grid point:

- (1) Normal stresses or strains in all three directions.
- (2) Shear stresses or strains in all three planes.
- (3) Principal stresses or strains in all three directions with associated direction cosines.
- (4) Mean stress or strain.
- (5) Octahedral shear stress or strain.



(a) ELEMENT GRID POINTS



(b) ELEMENT STRESS POINTS

Figure 7. IHEX1 Element Geometry

The stress and strain output at each of the nine points is identified by a stress or strain point ID. The stress and strain point IDs are numbered 1 through 9, with the first eight ordered as on the associated **CIHEX1** input data entry, and the ninth located at the element center, as illustrated in Figure 7. All output is provided in the basic coordinate system, since there is no naturally occurring element coordinate system for the IHEX1. An example of the output for the IHEX family of elements is shown in Table 28. The HEX1 element is shown with the HEX2 and HEX3 elements differing only in the number of data recovery points.

Strain energy output may be requested for the IHEX1 element. The strain energy print for the IHEX1 is identical to that for the BAR element and includes a breakdown by element and by element type.

Table 28. IHEX1 Element Solution Quantities

ELEMENT ID.	STRESS POINT	STRESS IN BASIC COORDINATE SYSTEM						ELEMENT TYPE		IHEX1	
		CENTER AND CORNER POINT STRESS		DIRECTIONAL STRESS		MEAN STRESS		TANGENTIAL STRESS		TANGENTIAL STRESS	
		NORMAL	SHEAR	AXIAL	TRANSVERSE	A	B	A	B	A	B
123	1	X -3.613902E+01	XY -2.852164E+01	A 6.42234E+00	IX 1.51 1.51 1.51	1.51 1.51 1.51	1.51 1.51 1.51	1.51 1.51 1.51	1.51 1.51 1.51	1.51 1.51 1.51	1.51 1.51 1.51
		Y -3.264816E+01	YZ 0.000000E+00	B 1.18650E+00	IY 1.51 1.51 1.51	1.51 1.51 1.51	1.51 1.51 1.51	1.51 1.51 1.51	1.51 1.51 1.51	1.51 1.51 1.51	1.51 1.51 1.51
		Z -3.264816E+01	ZX 0.108560E+01	C 1.18650E+00	IZ 1.51 1.51 1.51	1.51 1.51 1.51	1.51 1.51 1.51	1.51 1.51 1.51	1.51 1.51 1.51	1.51 1.51 1.51	1.51 1.51 1.51
123	2	X 7.238755E+01	XY 2.909098E+01	A 1.15170E+01	IX 1.44 1.44 1.44	1.44 1.44 1.44	1.44 1.44 1.44	1.44 1.44 1.44	1.44 1.44 1.44	1.44 1.44 1.44	1.44 1.44 1.44
		Y 3.132424E+01	YZ 0.000000E+00	B 1.87319E+00	IY 1.44 1.44 1.44	1.44 1.44 1.44	1.44 1.44 1.44	1.44 1.44 1.44	1.44 1.44 1.44	1.44 1.44 1.44	1.44 1.44 1.44
		Z 3.132424E+01	ZX 3.477172E+01	C 3.10217E+01	IZ 1.41 1.41 1.41	1.41 1.41 1.41	1.41 1.41 1.41	1.41 1.41 1.41	1.41 1.41 1.41	1.41 1.41 1.41	1.41 1.41 1.41
123	9	X 0.000000E+00	XY 0.000000E+00	A 0.00000E+00	IX 1.41 1.41 1.41	1.41 1.41 1.41	1.41 1.41 1.41	1.41 1.41 1.41	1.41 1.41 1.41	1.41 1.41 1.41	1.41 1.41 1.41
		Y 0.217932E+00	YZ 0.000000E+00	B 1.00000E+00	IY 1.41 1.41 1.41	1.41 1.41 1.41	1.41 1.41 1.41	1.41 1.41 1.41	1.41 1.41 1.41	1.41 1.41 1.41	1.41 1.41 1.41
		Z 0.504966E+01	ZX 0.000000E+00	C 0.44113E+01	IZ 1.41 1.41 1.41	1.41 1.41 1.41	1.41 1.41 1.41	1.41 1.41 1.41	1.41 1.41 1.41	1.41 1.41 1.41	1.41 1.41 1.41
ELEMENT ID.	STRAIN POINT	STRAIN IN BASIC COORDINATE SYSTEM						ELEMENT TYPE		IHEX1	
		CENTER AND CORNER POINT STRAIN		DIRECTIONAL STRAIN		MEAN STRAIN		TANGENTIAL STRAIN		TANGENTIAL STRAIN	
		NORMAL	SHEAR	AXIAL	TRANSVERSE	A	B	A	B	A	B
123	1	X 0.000000E+00	XY 0.000000E+00	A 0.00000E+00	IX 1.41 1.41 1.41	1.41 1.41 1.41	1.41 1.41 1.41	1.41 1.41 1.41	1.41 1.41 1.41	1.41 1.41 1.41	1.41 1.41 1.41
		Y 0.000000E+00	YZ 0.000000E+00	B 0.00000E+00	IY 1.41 1.41 1.41	1.41 1.41 1.41	1.41 1.41 1.41	1.41 1.41 1.41	1.41 1.41 1.41	1.41 1.41 1.41	1.41 1.41 1.41
		Z 0.000000E+00	ZX 0.000000E+00	C 0.00000E+00	IZ 1.41 1.41 1.41	1.41 1.41 1.41	1.41 1.41 1.41	1.41 1.41 1.41	1.41 1.41 1.41	1.41 1.41 1.41	1.41 1.41 1.41
123	2	X 0.000000E+00	XY 0.000000E+00	A 0.00000E+00	IX 1.41 1.41 1.41	1.41 1.41 1.41	1.41 1.41 1.41	1.41 1.41 1.41	1.41 1.41 1.41	1.41 1.41 1.41	1.41 1.41 1.41
		Y 0.000000E+00	YZ 0.000000E+00	B 0.00000E+00	IY 1.41 1.41 1.41	1.41 1.41 1.41	1.41 1.41 1.41	1.41 1.41 1.41	1.41 1.41 1.41	1.41 1.41 1.41	1.41 1.41 1.41
		Z 0.000000E+00	ZX 0.000000E+00	C 0.00000E+00	IZ 1.41 1.41 1.41	1.41 1.41 1.41	1.41 1.41 1.41	1.41 1.41 1.41	1.41 1.41 1.41	1.41 1.41 1.41	1.41 1.41 1.41
123	9	X 0.000000E+00	XY 0.000000E+00	A 0.00000E+00	IX 1.41 1.41 1.41	1.41 1.41 1.41	1.41 1.41 1.41	1.41 1.41 1.41	1.41 1.41 1.41	1.41 1.41 1.41	1.41 1.41 1.41
		Y 0.000000E+00	YZ 0.000000E+00	B 0.00000E+00	IY 1.41 1.41 1.41	1.41 1.41 1.41	1.41 1.41 1.41	1.41 1.41 1.41	1.41 1.41 1.41	1.41 1.41 1.41	1.41 1.41 1.41
		Z 0.000000E+00	ZX 0.000000E+00	C 0.00000E+00	IZ 1.41 1.41 1.41	1.41 1.41 1.41	1.41 1.41 1.41	1.41 1.41 1.41	1.41 1.41 1.41	1.41 1.41 1.41	1.41 1.41 1.41

5.2.1.5. IHEX2 Element Output

The IHEX2 element is a quadratic isoparametric solid hexahedron element with three extensional degrees of freedom for each of its twenty nodes.

Stresses, strains, and strain energies are available as output for the IHEX2 element through the **STRESS**, **STRAIN**, and **ENERGY** solution control print command options. Force output is not available for the IHEX2 element. On request, the following stresses and strains are output in the basic coordinate system at the twenty-one points located at the center, corners, and mid-edges of the element:

- (1) Normal stresses or strains in all three directions.
- (2) Shear stresses or strains in all three planes.
- (3) Principal stresses or strains in all three directions with associated direction cosines.
- (4) Mean stress or strain.
- (5) Octahedral shear stress or strain.

The stress and strain output at each of the twenty-one points is identified by a stress or strain point ID. The stress and strain point IDs are numbered 1 through 21, with the first twenty ordered as on the associated **CIHEX2** input data entry, and the twenty-first located at the element center. Although the corner stress and strain points are located at the corner grid points of the element, the mid-edge stress and strain points may or may not be located at the mid-edge grid points, depending on the location of those grid points. The stress/strain points for the IHEX2 are illustrated in Figure 8. All output occurring element coordinate system for the IHEX2.

Strain energy output may be requested for the IHEX2 element. The strain energy print for the IHEX2 is identical to that for the BAR element and includes a breakdown by element and by element type.

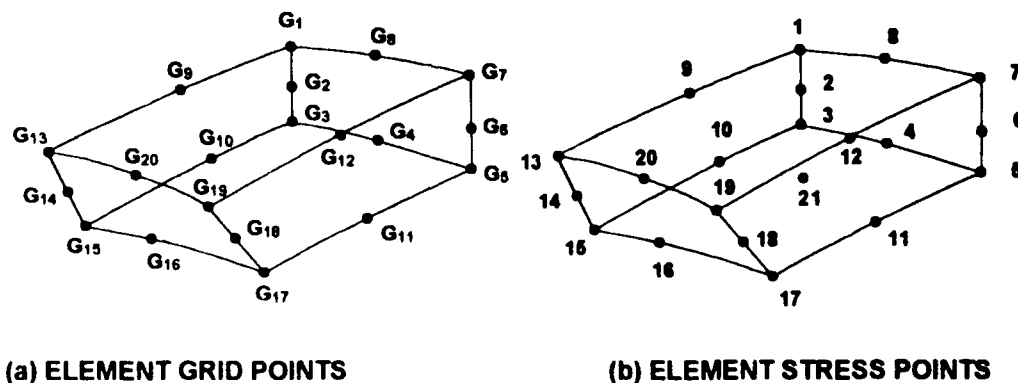


Figure 8. IHEX2 Element Geometry

5.2.1.6. IHEX3 Element Output

The IHEX3 element is a cubic isoparametric solid hexahedron element with three extensional degrees of freedom for each of its thirty two nodes.

Stresses, strains, and strain energies are available as output for the IHEX3 element through the **STRESS**, **STRAIN**, and **ENERGY** solution control print command options. Force output is not available for the IHEX3 element. On request, the following stresses and strains are output in the basic coordinate system at the twenty-one points located at the center, corners, and mid-edges of the element:

- (1) Normal stresses or strains in all three directions.
- (2) Shear stresses or strains in all three planes.
- (3) Principal stresses or strains in all three directions with associated direction cosines.
- (4) Mean stress or strain.
- (5) Octahedral shear stress or strain.

The stress and strain output at each of the twenty-one points is identified by a stress or strain point ID. The stress and strain point IDs are numbered 1 through 21. The first twenty points are ordered as on the associated **CIHEX3** input data entry, except that there is only one mid-edge point per edge, instead of two, and the twenty-first point is located at the element center. Although the corner stress and strain points are located at the corner grid points of the element, the mid-edge stress and strain points may or may not be located at a grid point, depending on the location of the mid-edge grid points. The stress/strain points for the IHEX3 are illustrated in Figure 9. All output is provided in the basic coordinate system, since there is no naturally occurring element coordinate system for the IHEX3.

Strain energy output may be requested for the IHEX3 element. The strain energy print for the IHEX3 is identical to that for the BAR element and includes a breakdown by element and by element type.

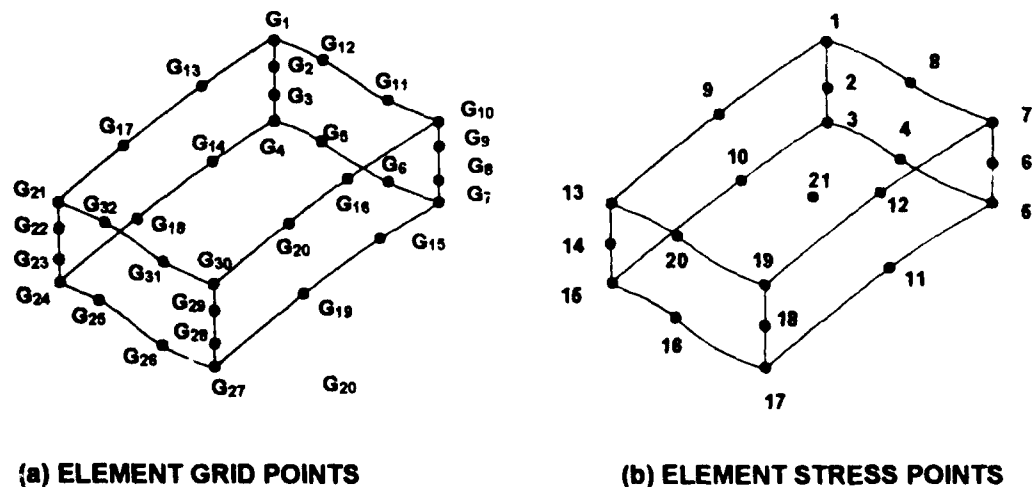


Figure 9. IHEX3 Element

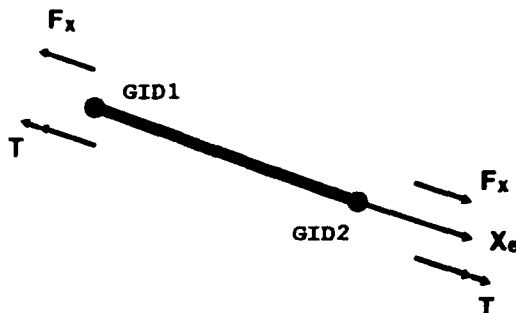


Figure 10. ROD Element Coordinate System

5.2.1.7. Rod Element Output

The ASTROS ROD element supports both extensional and rotational properties. The element coordinate system and sign conventions are shown in Figure 10. ASTROS supports stress, strain, force and strain energy output for the ROD. The forces that are computed are:

- (1) Axial force.
- (2) Torque about the element axis.

The torque and force are both computed even if the particular element does not support torsional or extensional forces, respectively. In these cases, a value of zero will be printed for the appropriate response quantity. The stresses and/or strains that are available are:

- (1) Axial stress or strain.
- (2) Torsional stress or strain.
- (3) Margin of safety for axial stress.
- (4) Margin of safety for torsional stress.

The margins of safety for strain are not available and the stress margins are computed even if there are no limits specified on the material property entry. In these cases, a large safety margin value is used to signify that no limits were imposed. An example of the ROD element output prints is shown in Table 29. The strain energy print for the ROD is identical to that for the BAR element and includes a breakdown by element and by element type.

5.2.1.8. QDMEM1/TRMEM Element Output

The QDMEM1 isoparametric element and the TRMEM constant strain triangular element are membrane elements which support isotropic, orthotropic and composite membrane properties. If the element is composite, the individual layers are treated as independent, stacked elements in which each "layer," as defined on the **PCOMP** bulk data entry, represents an element. In the case of composite elements, the layers are numbered sequentially starting with the first layer appearing on the **PCOMP** entry. Non-composite elements will show a layer number of zero.

Table 29. ROD Element Solution Quantities

STRESSES IN ROD ELEMENTS (GROUP 1)					STRESSES IN ROD ELEMENTS (GROUP 2)				
ELEMENT ID.	AXIAL STRESS	SAFETY MARGIN	TORSIONAL STRESS	SAFETY MARGIN	ELEMENT ID.	AXIAL STRESS	SAFETY MARGIN	TORSIONAL STRESS	SAFETY MARGIN
1	6.512166E+03	2.8E+00	0.000000E+00	1.7E+00	2	1.155248E+03	1.0E+01	0.000000E+00	1.0E+00
3	-6.821167E+03	2.7E+00	0.000000E+00	1.7E+00	4	1.155248E+03	1.0E+01	0.000000E+00	1.0E+00
1TEN BAR TRUSS					ASTROS VERSION 11.1 - ANALYSIS - P. 11				
FINAL STATIC ANALYSIS					FINAL ANALYSIS - ELEMENT				
STRAINS IN ROD ELEMENTS (GROUP 1)					STRAINS IN ROD ELEMENTS (GROUP 2)				
ELEMENT ID.	AXIAL STRAIN		TORSIONAL STRAIN		ELEMENT ID.	AXIAL STRAIN		TORSIONAL STRAIN	
1	6.512166E-04		0.000000E+00		2	1.155248E-04		0.000000E+00	
3	-6.821167E-04		0.000000E+00		4	1.155248E-04		0.000000E+00	
1TEN BAR TRUSS					ASTROS VERSION 11.1 - ANALYSIS - P. 12				
FINAL STATIC ANALYSIS					FINAL ANALYSIS - ELEMENT				
FORCES IN ROD ELEMENTS (GROUP 1)					FORCES IN ROD ELEMENTS (GROUP 2)				
ELEMENT ID.	AXIAL FORCE		TORQUE		ELEMENT ID.	AXIAL FORCE		TORQUE	
1	1.953650E+05		0.000000E+00		2	4.012462E+04		0.000000E+00	
3	-2.046350E+05		0.000000E+00		4	4.012462E+04		0.000000E+00	

Stresses, strains, forces and strain energies are available for each element or layer of a composite element. Since the stresses, strains, and forces vary within a QDMEM1 element, the intersection point of the diagonals projected onto the mean plane of a warped element has been chosen as the point at which the stresses, strains, forces and strain energies for the element are computed. The stresses, strains and element forces are computed in the element coordinate system. The element coordinate system and the stress computation point for the QDMEM1 element are shown in Figure 11 and those for the TRMEM in Figure 12.

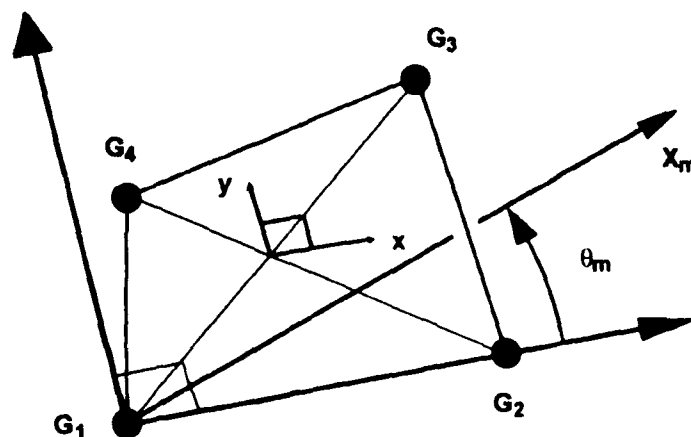


Figure 11. QDMEM1 Element Coordinate System

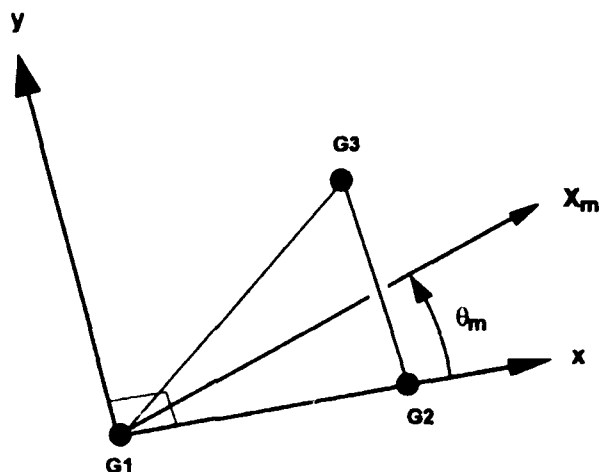


Figure 12. TRMEM Element Coordinate System

ASTROS computes the running loads associated with the stresses for the QDMEM1 element. These forces are:

- (1) The force components in the element coordinate system at the stress computation point.

The QDMEM1 stress and strain print includes the following:

- (2) The normal stresses or strains at the stress point in the element x- and y-directions.
- (3) The shear stress or strain on the element x face in the element y-direction.

Table 30. QDMEM1 Solution Quantities

SIMPLICIAL FLIGHTER WING										ASTROS VERSION 9.0 03/03/93 P. 13									
										FINAL ANALYSIS SEGMENT									
										STATICS ANALYSIS: BOUNDARY 1, SUBCASE 1									
STRESSES IN QUADRILATERAL MEMBRANES (QDMEM1)																			
ELEMENT	LAYER	STRESSES IN ELEMENT COORD SYSTEM				PRINCIPAL	PRINCIPAL STRESSES				MAX								
111	NO	NORMAL X	NORMAL Y	HEAR XY	STRESS ANGLE	MAJOR	MINOR	SHEAR											
1	0	5.124741E+03	2.00854E+03	1.106961E+03	16.4072	9.008202E+02	1.184507E+03	2.042664E+03											
2	0	5.512678E+03	2.12841E+03	1.256377E+03	18.0762	9.045232E+02	8.387216E+03	4.345870E+03											
3	0	6.414443E+03	2.114776E+03	1.165065E+03	74.0869	6.447088E+03	9.737824E+02	2.210435E+03											
SIMPLICIAL FLIGHTER WING										ASTROS VERSION 9.0 03/03/93 P. 15									
										FINAL ANALYSIS SEGMENT									
										STATICS ANALYSIS: BOUNDARY 1, SUBCASE 1									
STRAINS IN QUADRILATERAL MEMBRANES (QDMEM1)																			
ELEMENT	LAYER	STRAINS IN ELEMENT COORD SYSTEM				PRINCIPAL	PRINCIPAL STRAINS				MAX								
111	NO	NORMAL X	NORMAL Y	HEAR XY	STRAIN ANGLE	MAJOR	MINOR	SHEAR											
1	0	1.512474E-04	6.00854E-05	1.106961E-04	16.4072	1.095158E-04	1.481652E-04	5.433233E-04											
2	0	1.551267E-04	6.12841E-05	1.256377E-04	18.0762	1.071982E-04	8.487387E-04	1.155937E-03											
3	0	1.641444E-04	6.114776E-05	1.165065E-04	74.0869	6.768110E-04	2.111190E-04	5.879501E-04											
SIMPLICIAL FLIGHTER WING										ASTROS VERSION 9.0 03/03/93 P. 17									
										FINAL ANALYSIS SEGMENT									
										STATICS ANALYSIS: BOUNDARY 1, SUBCASE 1									
FORCES IN QUADRILATERAL MEMBRANES (QDMEM1)																			
ELEMENT	MEMBRANE																		
111	FX	FY	FGY																
1	1.149744E+02	1.117117E+02	2.213922E+02																
2	7.064506E+02	1.510085E+03	-5.127553E+02																
3	1.102066E+02	6.279500E+02	2.461330E+02																

- (4) The angle in degrees between the element x-axis and the major principal axis.
- (5) The major and minor principal (zero shear) stresses or strains.
- (6) The maximum shear stress or strain.

An example of the printed output for the QDMEM1 is shown in Table 30. The output for the TRMEM is identical except for the titling.

The strain energy print for the QDMEM1 is identical to that for the BAR element and includes a breakdown by element and by element type.

5.2.1.9. QUAD4/TRIA3 Element Output

The QUAD4 and TRIA4 isoparametric quadrilateral and triangular plate elements include both membrane and bending behavior. Transverse shear flexibility may be requested, as can the coupling of membrane and bending behavior. The QUAD4 element coordinate system and node numbering are shown in Figure 13. The TRIA3 element coordinate system and node numbering are shown in Figure 14. These elements may be assigned general anisotropic or composite material properties. For designed composites, the layers are treated as stacked membrane elements similar to the QDMEM1 element. In this case, the layers are identified by number in the order specified on the **PCOMP**, **PCOMP1** or **PCOMP2** entry. For design invariant composite laminates, the output always refers to the aggregate laminate properties and refers to layer number zero. The reference plane of the QUAD4/TRIA3 elements may be offset from the plane of the grid points and variation in the element thickness may be modeled by assigning different element thicknesses at each of the grid points. The reader is referred to Appendix A of the ASTROS Theoretical Manual for additional information on these plate bending elements.

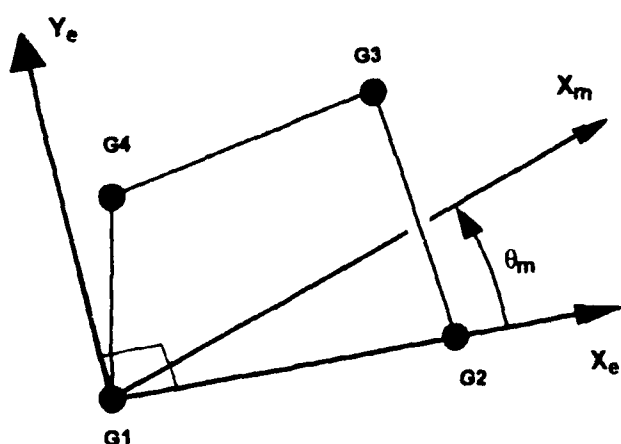


Figure 13. QUAD4 Element Coordinate System

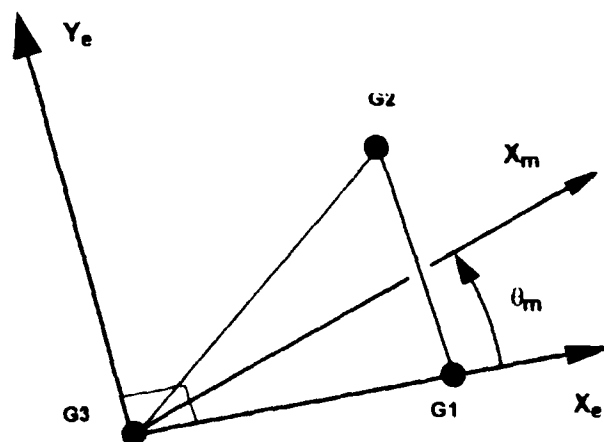


Figure 14. TRIA3 Element Coordinate System

Table 31. QUAD4 and TRIA3 Solution Quantities

SIMPLIFIED FIGHTER WING										ASTROS VERSION 9.0 03/03/93 P. 13	
										FINAL ANALYSIS SEGMENT	
										STATICS ANALYSIS: BOUNDARY 1, SUBCASE 1	
STRESSES IN QUADRILATERAL PLATES (QUAD4)											
ELEMENT	LAYER	FIBER	STRESSES IN STRESS COORD SYSTEM			PRINCIPAL STRESSES (ZERO SHEAR)					
ID	NO.	DISTANCE	NORMAL X	NORMAL Y	SHEAR-XY	ANGLE	MAJOR	MINOR			
5	0	1.00000E-01	6.91372E+02	8.33210E+03	2.21727E+03	-13.0858	1.20677E+03	-8.84749E+03			
		1.00000E-01	6.91372E+02	-8.33210E+03	-2.21727E+03	-13.0858	1.20677E+03	-8.84749E+03			
		1.00000E-01	5.43684E+02	-7.64178E+03	2.58719E+03	-18.0457	2.99228E+02	-8.48469E+03			
		1.00000E-01	5.43683E+02	-7.64178E+03	2.58719E+03	-18.0457	2.99228E+02	-8.48469E+03			
10	0	1.00000E-01	6.16923E+02	3.07066E+03	1.17650E+03	73.7292	3.41404E+03	-9.60305E+02			
		1.00000E-01	6.16923E+02	3.07066E+03	1.17650E+03	73.7292	3.41404E+03	-9.60305E+02			
SIMPLIFIED FIGHTER WING										ASTROS VERSION 9.0 03/03/93 P. 14	
										FINAL ANALYSIS SEGMENT	
										STATICS ANALYSIS: BOUNDARY 1, SUBCASE 1	
STRAINS IN QUADRILATERAL PLATES (QUAD4)											
ELEMENT	LAYER	FIBER	STRAINS IN STRESS COORD SYSTEM			PRINCIPAL STRAINS (ZERO SHEAR)					
ID	NO.	DISTANCE	NORMAL X	NORMAL Y	SHEAR-XY	ANGLE	MAJOR	MINOR			
5	0	1.00000E-01	3.50886E-04	8.56139E-04	-5.91205E-04	-13.0479	4.19391E-04	-9.24645E-04			
		1.00000E-01	3.50886E-04	8.56139E-04	-5.91205E-04	-13.0479	4.19391E-04	-9.24645E-04			
		1.00000E-01	1.94048E-04	7.46862E-04	6.85105E-04	-18.0443	3.04642E-04	-8.58457E-04			
		1.00000E-01	1.94048E-04	7.46862E-04	6.85105E-04	-18.0443	3.04642E-04	-8.58457E-04			
10	0	1.00000E-01	1.64593E-04	3.26487E-04	3.13462E-04	73.7247	3.72245E-04	-2.10351E-04			
		1.00000E-01	1.64593E-04	3.26487E-04	3.13462E-04	73.7247	3.72245E-04	-2.10351E-04			
SIMPLIFIED FIGHTER WING										ASTROS VERSION 9.0 03/03/93 P. 15	
										FINAL ANALYSIS SEGMENT	
										STATICS ANALYSIS: BOUNDARY 1, SUBCASE 1	
FORCES IN QUADRILATERAL PLATES (QUAD4)											
ELEMENT	MEMBRANE FORCES		BENDING MOMENTS		-TRANSVERSE SHEAR FORCES-						
ID	FX	FY	FX	MY	QX	QY					
5	2.10845E+02	1.74879E+04	2.46591E+02	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00		
7	5.94785E+01	1.69688E+03	1.03084E+01	2.03451E-07	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00		
10	1.89788E+02	6.47546E+02	4.45474E+01	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00		

Stresses, strains, forces, and strain energies are available as output for the QUAD4/TRIA3 elements through the **STRESS**, **STRAIN**, **FORCE** and **ENERGY** solution control print command options. The following element stresses and strains are output on request:

- (1) Combined extensional and bending stresses and strains computed at the element center in the element coordinate system.
- (2) Principal stresses and strains computed at the element center including the angle between the element x-axis and the principal axis.

The following forces are output on request:

- (1) Element forces computed at the center of the element in the mean plane in the element coordinate system.

For composite materials, all output quantities are computed using the aggregate laminate properties. Hence, output of stresses or strains at the ply or laminae level is currently not an available print option for the QUAD4/TRIA3 elements in ASTROS. An example of the printed output is shown in Table 31.

5.2.1.10. Shear Panel Output

The shear panel is an element which resists the action of tangential forces applied to its edges. In ASTROS, the shear panel supports only isotropic material properties and makes use of the shear flow distribution approximation of Garvey (Reference 4) with special handling for warped, parallel edge and general trapezoidal geometries. The element force sign convention is shown in Figure 15. The stresses, strains, forces and strain energies are available for the shear panel. The element forces that are computed include the following:

- (1) The eight forces between each pair of nodes; each force is directed along the line connecting the adjacent nodes (the element edge).
- (2) The four "kick" forces at each node, normal to the plane formed by the two adjacent element edges.
- (3) The shear flows (forces/unit length) along each edge.

When stresses or strains are requested, they are computed at the rule points in skewed coordinates parallel to the adjacent edges. Both the average and maximum shear stress or strain are then printed. A safety margin based on the maximum stress value is computed for stress output. A large safety margin is printed if not limits were specified on the material property entry. Table 32 contains a sample of the SHEAR panel element output.

The strain energy print for the SHEAR panel is identical to that for the BAR element and includes a breakdown by element and by element type.

5.2.2. Nodal Response Quantities

ASTROS has two basic forms of node point: the structural node and the extra point. The structural node is defined as either a "grid" point having six degrees of freedom (three translations and three rotations) or a "scalar" point having a single degree of freedom. These node points can be used to connect metric and scalar structural elements. The extra point is similar to the scalar point in that it has a single degree of freedom, but differs in that extra points included in the model are selected in the boundary condition rather than being implicitly included in the model. Further, they cannot be connected directly to either metric or scalar structural elements; instead, these elements are connected through terms introduced by direct matrix input or by transfer functions. Extra points are used in dynamic analyses for modeling control systems and other nonstructural mechanisms in the system under analysis. These degrees of freedom do not appear in the system matrices until after the dynamic matrix assembly and do not appear in any but the dynamic response disciplines (**FLUTTER**, **TRANSIENT**, **FREQUENCY** and **BLAST**). When nodal output is requested for dynamic analyses, any extra point results may be selected using the **GRIDLIST** entry just as are grid and scalar point results.

Nodal output is available for all disciplines in ASTROS, although particular nodal response quantities may not be available for all disciplines. The solution control print options **VELOCITY**, **DISPLACEMENT**, **GPFORCE**, **LOAD**, **SPCFORCE**, and **ACCELERATION** are used to select print of the nodal response quantities. Each of these print options selects either **ALL**, **NONE** or an integer set identification number that refers to one or more **GRIDLIST** bulk data entries. Chapter 3 contains the complete description of the solution control print command. Each output is carefully labeled as to its boundary condition number, which discipline generated the output quantities and which load condition, mode shape, time step, frequency step or flight condition is represented by the output.